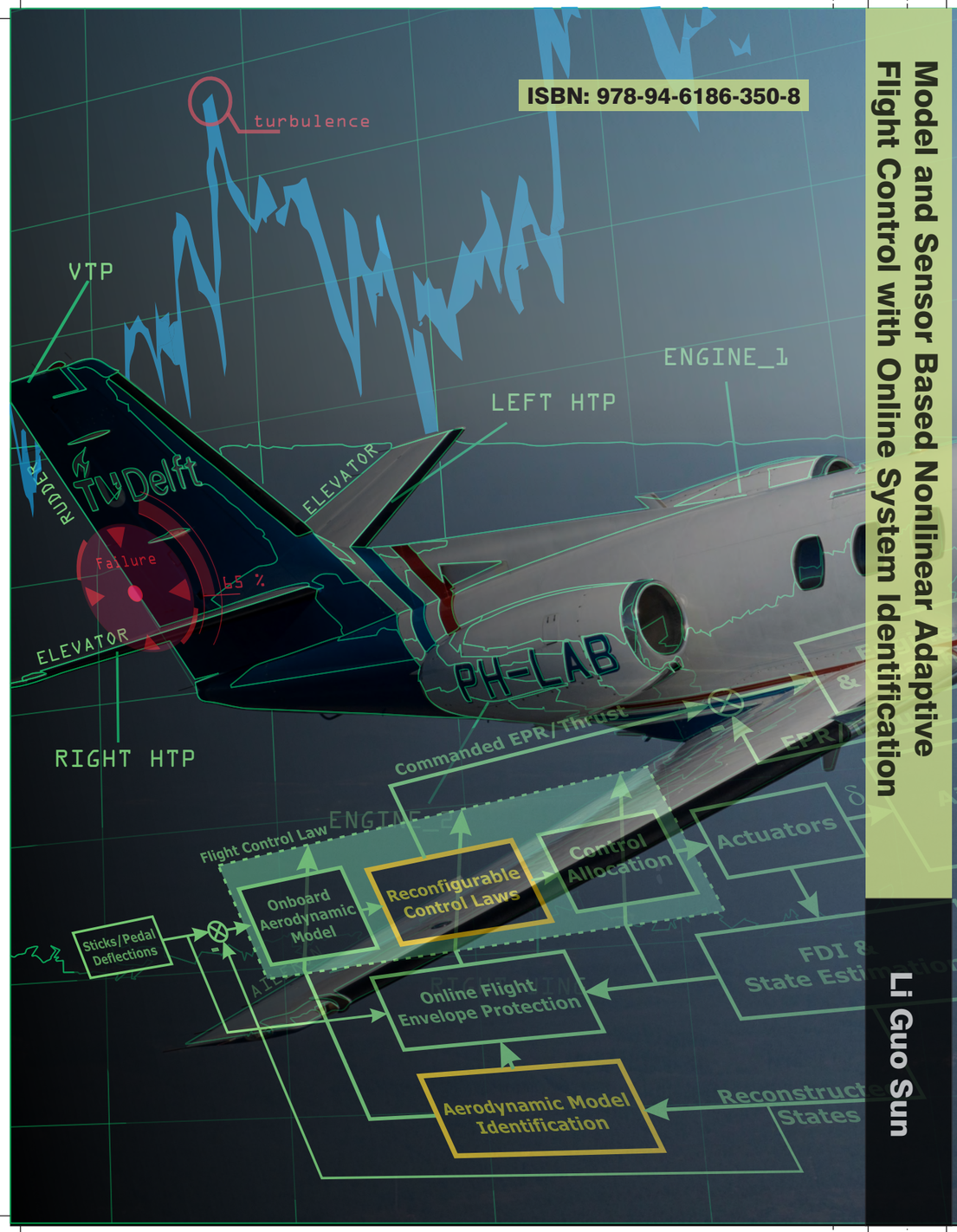
ISBN: 978-94-6186-350-8



Model and Sensor Based Nonlinear Adaptive Flight Control with Online System Identification

Li Guo Sun

Model and Sensor Based Nonlinear Adaptive Flight Control with Online System Identification

Li Guo Sun
孙立国

Invitation

On thursday
October 30, 2014
at 12.30

Li Guo SUN

will defend his thesis titled:

Model and Sensor Based Nonlinear Adaptive Flight Control with Online System Identification

The defense will take place in the Senaatszaal at the Aula of TU Delft Mekelweg 5, Delft

At 12.00, prior to the defense, there will be a short presentation about the thesis work

MODEL AND SENSOR BASED NONLINEAR ADAPTIVE FLIGHT CONTROL WITH ONLINE SYSTEM IDENTIFICATION

Li Guo SUN

基于在线辨识模型和传感器的非线性自适应飞行控制

孙立国

MODEL AND SENSOR BASED NONLINEAR ADAPTIVE FLIGHT CONTROL WITH ONLINE SYSTEM IDENTIFICATION

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 30 oktober 2014 om 12.30 uur

door

Li Guo SUN

Master of Science, Nanjing University of Aeronautics and Astronautics
geboren te Tangshan, Hebei Province, China

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. M. Mulder

Copromotor: Dr. Q. P. Chu

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. M. Mulder,	Technische Universiteit Delft, promotor
Dr. Q.P. Chu,	Technische Universiteit Delft, copromotor
Prof.dr.ir. M. Verhaegen,	Technische Universiteit Delft
Prof.dr.-lng F. Holzapfel,	Technische Universität München
Dr.ir. G.H.N. Looye,	Deutsches Zentrum für Luft and Raumfahrt (DLR)
Prof.dr. A. Zolghadri,	University of Bordeaux
Dr.ir. C.C. de Visser,	Technische Universiteit Delft
Prof.dr.ir. J.A. Mulder,	Technische Universiteit Delft, reservelid

Keywords: Fly-by-wire, Aerodynamic model, Adaptive control, Fault tolerant, Reconfiguration, Flight envelope protection, Simplex spline theory

Printed by: Ipskamp Drukkers, Enschede, The Netherlands.

Cover photo copyright © Ben Ullings, Aviation Photos International.
Cover design by Yazdi Ibrahim Jenie.

ISBN 978-94-6186-350-8

Published and distributed by L. G. Sun.
Email: sunliguo963@gmail.com

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To my parents and my wife

SUMMARY

Model and Sensor Based Nonlinear Adaptive Flight Control with Online System Identification

Li Guo Sun

Consensus exists that many loss-of-control (LOC) in flight accidents caused by severe aircraft damage or system failure could be prevented if flight performance could be recovered using the valid and remaining control authorities. However, the safe maneuverability of a post-failure aircraft will inevitably be reduced due to the malfunction. Non-conventional control strategies which rely on modern control techniques and computational power are essential to control systems in post-failure flight conditions to extract the most from the reduced, remaining aircraft control authorities and restore the flight performance of an aircraft or achieve a safe landing. One such non-conventional control strategy is called active fault tolerant flight control (FTFC), which is designed to detect changes in an aircraft's dynamics caused by structural, actuator, or sensor failure and accommodate the damage or failure using an adaptive reconfiguration mechanism. The active FTFC technique is able to deal with unanticipated and multiple simultaneous failures.

The overall architecture of an active FTFC system ideally should consist of a fault detection and diagnosis (FDD) module, a state reconstruction unit, a reconfigurable control component, a control allocation unit and a flight envelope protection (FEP) unit. Generally speaking, FTFC systems can be classified into two types: model-based FTFC systems and model-free FTFC systems, according to whether any of the system's components require an aerodynamic model at their core or not. A model-based FTFC system contains an aerodynamic model identification (AMI) module, which supplies an accurate aircraft model to an indirect adaptive nonlinear controller in the reconfigurable control block, to a dynamic flight envelope determination algorithm in an FEP unit, or to an FDD unit. An aerodynamic model identification approach using a physical, interpretable modeling structure can detect and even quantify structural failures occurring in the aircraft structure or one of the control surfaces by monitoring changes in stability derivatives and control derivatives.

There are many candidate control approaches which can achieve reconfiguration when designing a reconfigurable flight controller. These reconfigurable control methods may rely on many different reconfiguration mechanisms ranging from switching, model following, matching to adaptive compensation. These methods include nonlinear adaptive control which achieves reconfiguration through compensation, and this method is receiving increasing attention in the flight control aerospace research community. Nonlinear adaptive control is divided into direct adaptive control and indirect adaptive

control, the difference is that the latter requires an online system model. Indirect adaptive control is also called model-based or modular adaptive control, which has some advantages over the direct adaptive control and other model-free control methods. One advantage is that a modular control approach has the potential to yield a more efficient controller which requires less control effort. Such an efficient controller can be achieved by maintaining useful damping terms of an identified system model in the closed-loop system. This is attributed to the good properties of many control design techniques such as backstepping such that the dynamics of an original system can be chosen to be canceled or maintained during a controller design process. Modular adaptive control also has an inherited shortcoming, it can only guarantee *input-to-state* stability, i.e. modular adaptive control cannot guarantee the stability of the overall closed-loop system because its stability proof relies on the *certainty equivalence* principle. The weakness of the *certainty equivalence* principle, i.e. convergence problem of the model parameters, can be improved by enhancing model accuracy or reliability, to do this, it becomes critical to develop advanced, powerful aerodynamic model identification approaches capable of capturing changes in flight dynamics either during a high maneuvering flight mission or a post-failure condition.

Flight envelope protection is a necessary technique that should be applied by controller designers to prevent LOC incidents, taking into account highly maneuvering flight tasks and/or highly perturbed flight conditions due to the ongoing failure. An FEP component should provide a pilot with a safe flight envelope and pose constraints on the reference commands fed to an internal controller to make the commands achievable.

An aerodynamic model that is valid over an entire flight envelope plays a crucial role in full-envelope modular adaptive control and flight envelope protection. A globally valid model is required for modular adaptive control to enable the designed controller to work properly in a large operating range. Once estimated, the global model in a model-based adaptive control method can be stored for later re-use when the same flight condition is revisited. Except being needed by a model-based controller, an accurate aerodynamic model is also required for flight envelope protection. Naturally, the estimated aerodynamic model has to be valid for the current aircraft configuration over the entire flight envelope to enable an evolution algorithm to estimate the boundary of the safe flight envelope for the current flight condition. However, only a limited number of model identification approaches are suited for estimating a globally valid aerodynamic model, and each existing possible candidate has variant shortcomings or limitations which make it hard to apply directly to identify an aircraft model. For example, neural networks usually yield a nontransparent model structure which is hard to interpret using physical knowledge of the system, and they commonly encounter a convergence problem. Most kernel methods fall into the nonparametric type of methods, which by nature need as many kernels as the data points under evaluation. It should be kept in mind that only *equation-error* type model identification methods were investigated in the work reported here. The assumption was made that a sufficiently accurate estimation of aircraft states was available.

An alternate method to the modular adaptive reconfigurable control approach is the acceleration measurements-based incremental nonlinear control (AMINC) method. An accurate estimation of an aircraft is hard to achieve during a high maneuvering moment

or at a transient period when the flight performance is highly perturbed due to aircraft failure. Incremental nonlinear controllers such as incremental nonlinear dynamic inversion (INDI), incremental backstepping (IBKS) and sensor-based backstepping (SBB) are suited for reconfigurable flight control designs in the sense that they do not require complete aircraft model knowledge.

The main research question for the research presented here was: How can an advanced fault-tolerant flight control system be designed to increase the survivability of an aircraft? This led to two subsidiary questions:

- How can the candidate function approximation methods, i.e. multivariate simplex B-splines and kernel methods, be improved in terms of approximation accuracy and computational efficiency, to meet the need of model-based adaptive control and online flight envelope protection?
- What are the benefits of using an acceleration measurements-based control approach, i.e. the sensor based backstepping, as an alternative to a model-based adaptive control approach, when designing a reconfigurable flight controller to deal with aircraft failures in a generic fault-tolerant flight control (FTFC) system?

With regard to reconfigurable control, the identified model should enable the controller to achieve active reconfiguration and restore the control performance. To answer these questions, four different global model identification methods and two nonlinear incremental adaptive controllers were developed.

Two model identification methods use a parametric model structure namely standard multivariate simplex B-splines. The focus was placed on how to achieve fast parameter estimation during the research process for these two methods. In the third identification method, a new model structure called tensor-product simplex B-splines was extended from a single dimension case to a multidimensional case, with a focus on demonstrating the advantage of this new compound model structure in terms of the flexibility in model structure selection, computational efficiency and approximation power. The fourth method uses a kernel type model structure which is also parametric. The new recursive kernel approach was developed by combining a classical recursive kernel method with a novel support vector regression approach.

A model identification method using standard multivariate simplex B-splines has many advantages, it can avoid the over-fitting problem which occurs with an ordinary polynomial method using a triangulation technique. The approximation power of a simplex B-spline based method is determined by the per-simplex polynomial order and smoothness order, and can be increased by increasing the density of the subdomains in a triangulation. This simplex B-spline based function approximation method guarantees that its output is bounded by the maximum and minimum B-coefficients, this facilitates its certification for future real life applications. The linear regression formulation of the simplex B-spline based method allows for applying most of the constrained recursive parameter estimation methods. Furthermore, the simplex B-spline based method has a sparse property, which can lead to high computational efficiency by adopting distributed computation or other modern computing techniques. However, a simplex B-spline method can easily yield a large amount of unknown parameters if the function dimension exceeds 4, which results in a high computational load considering the smoothness maintaining and covariance matrix updating.

To enhance the computational efficiency of the model identification methods using simplex B-splines, two recursive linear-regression model identification methods were developed in this thesis: a substitution-based multivariate simplex B-spline (SB-MVSB) method and a recursive sequential multivariate simplex B-spline (RS-MVSB) method. In the SB-MVSB method, an efficient recursive solver is developed for a constrained linear regression problem when using simplex B-splines. The constrained linear regression problem is converted into a constraint-free linear regression problem using a general solution for the equality constraints. This transformation was shown to reduce the scale of the identification problem in terms of the number of unknown parameters, and thus the computational load required for the model identification method can be reduced.

The RS-MVSB method consists of two consecutive procedures at one model evolution step. The first procedure achieves updating of a local model covering the current data point instead of a global model. The requirement of updating a complete covariance matrix is avoided by only updating one local model, and therefore the computational efficiency of this method is greatly enhanced. The second procedure guarantees a smooth transition between this local model and its neighboring local models.

The computational complexity of SB-MVSB and RS-MVSB was given from a mathematician point of view, then, they were validated using simulated flight test data generated using a high-fidelity nonlinear model of an F-16 aircraft. Simulation results showed that both methods can achieve higher approximation accuracy than ordinary polynomial based methods, and both can be many, e.g. 10, times faster than an equality constraint recursive least squares based MVSB (ECRLS-MVSB) method. The second feature of these two methods facilitates their future onboard applications.

Tensor-product simplex (TPS) B-splines provide a compound structure, which provide more flexibility than a standard simplex B-spline model during model structure selection. Using TPS B-splines, different dimension of inputs can be treated differently depending on their characteristics determined from a priori knowledge. In the work presented in this thesis, the TPS B-spline concept was extended from a single dimension case into a more general multidimensional case. Compared to standard simplex B-splines, TPS B-splines can make better use of a priori model knowledge. By reducing many unnecessary basis polynomials from the regression vector, TPS B-splines have the potential to lead to a lower computational load than standard simplex B-splines. The TPS B-spline method was validated using a data set generated from a high-fidelity nonlinear F-16 model. Simulation results showed that TPS B-splines can yield higher approximation power than standard simplex B-splines with less B-coefficients.

Two similar recursive parametric kernel methods namely weight varying least squares support vector regression (WV-LSSVR) and Gaussian process kernel based LSSVR (GPK-LSSVR) were developed for aerodynamic model identification in this thesis. The focus of this work was enhancing the approximation power of a recursive parametric kernel method by choosing an optimal set of kernels for the kernel scheme. An offline method called improved recursive reduced LSSVR (IRR-LSSVR) was used to determine optimal kernels for a classical recursive kernel method. The new kernel method was validated using a series of public available benchmark data sets well known to researchers from the field of pattern recognition. GPK-LSSVR showed a higher approximation power than

WV-LSSVR, and both of them showed a higher approximation power than a classical recursive kernel method based on *k-means clustering*.

A novel type of acceleration measurements-based incremental flight control laws was investigated with the aim of providing a reconfigurable control unit with a powerful non-conventional flight control approach which could accommodate sudden structural or actuator failures occurring in an aircraft. The preferred model-free, incremental control approach used in this thesis was the SBB approach, which was initially developed for control designs of nonlinear nonaffine-in-control systems. The SBB approach achieves an accurate reference command tracking performance by approximate dynamic inversion. The SBB approach was extended to deal with sudden model changes in an aircraft caused by structural or actuator failures. A hybrid two-loop angular controller and a joint two-loop angular controller were designed for the RECOVER model. In the hybrid two-loop angular controller, the angular control loop was designed using a nonlinear dynamic inversion (NDI) control law, and the angular rate loop controller using the SBB approach. In the joint two-loop angular controller, the overall controller was designed using a backstepping technique with each loop stabilized recursively. Both angular controllers were validated using the RECOVER model with a focus on dealing with perturbed aircraft flight performance caused by failures. Two benchmark fault scenarios were selected: a rudder runaway case and a flight 1862 engine separation scenario. Simulation results showed that both control setups can guarantee the safety of the post-failure aircraft and achieve a proper reference tracking performance. In comparison with the hybrid NDI/SBB angular controller, the joint SBB angular controller resulted in a better reference tracking performance for the sideslip angle, especially in the engine separation case.

An SBB controller contains a time scale parameter, other incremental control laws such as incremental NDI (INDI) and incremental backstepping (IBKS) involve a control effectiveness matrix. Before we can investigate how the time scale parameter or a control effectiveness matrix affect the control performance of an incremental flight controller, the parameter variations of a control effectiveness matrix need to be estimated and analyzed. The TPS B-spline method and an immersion and invariance (I&I) method were chosen to estimate a control effectiveness matrix for an F-16 aircraft. Although the I&I approach initially was not aimed at high modeling accuracy, it was assumed in this thesis that it is able to estimate the changing trend of the control derivatives. Simulation results showed that TPS B-splines capture the changes in the control derivatives better than the I&I approach in terms of consistency. For F-16, the control effectiveness matrix does not evidently affect the control performance of an incremental flight controller when a flight maneuver is moderate in terms of the variation of angle of attack and airspeed.

Further research on modular adaptive reconfigurable control is required, for example incorporating the SB-MVSB method or the WV-LSSVR method into control designs to further check how well they are suited for modular adaptive control in terms of approximation power and onboard computational efficiency. Further research on acceleration measurements based reconfigurable control should include tests on the SIMONA simulator, realistic test-flight with UAV and research aircraft.

CONTENTS

ACRONYMS

ADS	air data sensor 20
AMI	aerodynamic model identification 16
ANDI	adaptive nonlinear dynamic inversion 173
IMU	inertial measurement unit 20
EKF	extended Kalman filter 188
FDD	fault detection and diagnosis 15
FDI	fault detection and isolation 3
FEP	flight envelope protection 1
OFEP	online flight envelope protection 3
SFEP	static flight envelope protection 5
FTFC	fault tolerant flight control 1
HJB	Hamilton-Jacobi-Bellman 6
MVSB	multivariate simplex B-splines 7
NDI	nonlinear dynamic inversion 8
RBF-NN	radial basis functions neural networks 27
RFC	reconfigurable flight control 21
RMSE	root mean squared errors 40
RR-LSSVR	recursive reduced least squares support vector regression 171
SBB	sensor-based backstepping 11
SPT	singular perturbation theory 87
TP-MVSB	tensor-product multivariate simplex B-splines 170
TA	tuning algorithm 95
IRR-LSSVR	improved recursive reduced least squares support vector regression 67

GREEK SYMBOLS

ρ	=	density of air, kg/m^3
α,β	=	angle of attack, angle of sideslip, rad
ϕ,θ,ψ	=	roll angle, pitch angle, yawing angle, degree
ν	=	virtual control input
δ_a,δ_r	=	aileron and rudder deflections, degree
δ_e,δ_{lef}	=	elevator deflection, deflection of leading edge flap, degree
$\mathcal{T}$	=	triangulation
κ	=	a multi-index on barycentric coordinates
χ	=	heading angle, degree
γ	=	flight path angle, degree
ξ	=	residual vector
ϵ	=	time scale parameter for SBB controller

LATIN SYMBOLS

b	=	wing span, m
$b(\mathbf{x})$	=	barycentric coordinate vector of point $\mathbf{x}$
C_l,C_m,C_n	=	dimensionless roll, pitch, and yaw moment coefficient
C_X,C_Y,C_Z	=	dimensionless force coefficient in the body X, Y, and Z direction
$\bar{c}$	=	mean aerodynamic chord, m
$\mathbf{H}$	=	smoothness constraint matrix in simplex B-splines
$\mathbf{I}$	=	moment of inertia matrix
J	=	total number of simplices
$\boldsymbol{J}$	=	cost function
he, xe, ye	=	ground altitude, X-position and Y-position, m
P_c, P_d	=	collective and differential engine pressure ratio
$\mathbf{P_t}$	=	a vector of total engine pressure ratios
p,q,r	=	roll, pitch and yaw rate around the body axis, rad/s
S	=	wing area, m^2
$\mathbf{u}$	=	input vector
V_{TAS}	=	true airspeed, m/s
$\mathbf{x}$	=	state vector
$\mathbf{Y}$	=	observation vector

SUBSCRIPTS

ir, il, or, ol	=	acronym of inboard right (wing), inboard left (wing), outboard right, outboard left
r	=	reference command

1

INTRODUCTION

1.1. FLIGHT SAFETY AND THE NEED FOR RESILIENCE

The use of mechanical and later electronic aids for controlling flight in aircraft has a long history, almost as long as that of powered flight [90, 111]. The development of flight control systems has seen three stages. In early aircraft, cables, pulleys, levers and other mechanical devices were used to assist the pilot. Then hydro-mechanical systems were developed and finally today we have fly-by-wire systems. Each flight control system grew out of the previous system, with small aircraft mechanical systems were possible and sufficient, but as aircraft increased in size and the air speed and safety became paramount with the advent of mass commercial travel, the need grew for assisted hydraulic-mechanical systems. Finally, with the airbus A320 airliner we moved to completely fly-by-wire systems. In a parallel process, fighter aircraft developed from comparatively simple aircraft, pilot controlled, such as the Spitfire and the Messerschmitt through to today's sophisticated, fast and agile aircraft that require equally sophisticated, fast and agile flight control systems to aid the pilots flying them.

The initial concept of fault tolerant flight control (FTFC) can be traced back to the 1950s [130], when the United Air Force carried out a series of flight tests on adaptive flight control designs for fighter aircraft. These investigations were aimed at making these control systems tolerant of uncertainties and at the same time design control systems that would not need gain-scheduling. In the early 1990, Airbus started to use FTFC techniques in the Airbus A340 aircraft to enhance safety, in particular, attention was paid to making full use of redundant components, e.g., control surfaces, sensors, actuators.

Safety is critical when designing and operating civil aircraft because they carry passengers and fly to and from areas that are densely polulated. To increase safety, modern civil aircraft such as the Boeing 777 and Airbus A380, see Figs 1.1-1.2, are equipped with flight envelope protection (FEP) systems that prevent these aircraft from entering upset flight conditions. Existing FEP units contain a set of pre-defined logic that is used to regulate the reference commands of a controller such as bank angle ϕ, angle of attack α, and true airspeed V_{TAS}. These aircraft control systems

can be further improved by incorporating more advanced flight envelope protection approaches including model-based static and dynamic estimation of the safe flight envelope [19, 52, 54, 66, 85, 149, 164].

Figure 1.1: KLM Boeing 777-206/ER PH-BQD, by Tommy Desmet, via *airliners.net*.

Figure 1.2: Airbus A380, available at http://www.airbus.com/galleries/photo-gallery.

A high level of survivability is required when designing a flight control system for the new generation of fighter aircraft. Aerodynamic designs of fighter aircraft tend to be increasingly complex because these aircraft are required to have higher and higher levels of flight performance, e.g., maneuverability. Due to the increased complexity of the aerodynamic design and the need to fly multiple combat missions in harsh conditions, e.g., to launch missiles and fly in desert conditions, a fighter aircraft is more likely to encounter sudden model changes, e.g., structural or actuator damages, than a civilian aircraft. To obtain a high level of flight performance, a powerful control method is required for the flight control of a fighter aircraft. To enable a fighter aircraft to survive sudden model changes, the flight control system should either be robust to the model changes or be able to adapt to the changes.

Looking at the statistics for recent airliner accidents and incidents, those categorized as 'loss of control in flight' cover 23% of all aircraft accidents [18, 116], and it is this category that has recently received the most attention from industry. Some of these types of flight accidents can be avoided using the technology and computational power available at this moment [12, 85, 131]. Research into previous flight accidents [131] and investigations of the FTFC strategies used suggests that an aircraft, under many post-failure circumstances, can still achieve a certain level of flight performance using the remaining valid control effectors [131]. However, the control authority or the safe flight envelope of the aircraft will inevitably shrink due to structural/actuator failures. Therefore, to avoid the aforementioned type of flight accidents, it is necessary to employ suitable non-conventional control strategies to make the best possible use of the remaining flight potential from a post-failure aircraft [85, 131]. Amongst other initiatives, this is supported by the work of the Flight Mechanics Action Group 16 (FM-AG16) group, a branch of the Group for Aeronautical Research and Technology in Europe (GARTEUR).

1.2. FACTORS IN FAULT TOLERANT FLIGHT CONTROL

The use of FTFC has attracted a large amount of interest from the aerospace community. Enhancing the survivability of an aircraft during an incident is the ultimate goal when using or designing an FTFC system. In this thesis, a generic FTFC system is assumed to have the following six basic components: fault detection & isolation, state estimation, aerodynamic model identification, flight envelope protection, flight control law, and control allocation, see Figure 1.3.

An FTFC system should have the following functions: once structural or actuator failures occur in an aircraft, the FTFC system is expected to be able to, first, detect the failure, second, determine the safe-flight-envelope, and third, prevent the aircraft from entering any unsafe regions of the flight envelope by using reconfigurable control techniques. Using the failure knowledge obtained by a fault detection and isolation (FDI) unit, an online flight envelope protection (OFEP) unit will estimate a maneuvering safe-flight envelope in real time. The situation awareness of the pilots can be improved by incorporating a safe-flight-envelope indicator on the display interface to support their decision-making [12, 85]. The detected failure information obtained by an FDI unit also goes into the reconfigurable flight control unit, this allows the latter to perform reconfigurations during control allocation.

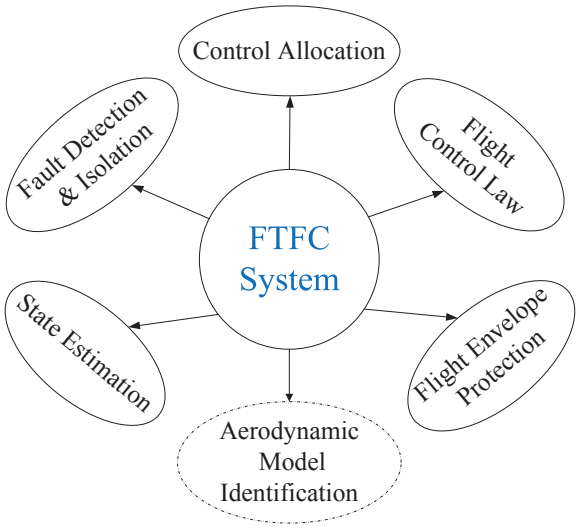


Figure 1.3: The components included in an FTFC system. (Note that the AMI block in the dashed ellipse is required only for a model-based FTFC system.)

There are two types of FTFC systems: model-free FTFC systems and model-based FTFC systems [86, 175]. In this thesis, a model-based FTFC system is defined as follows.

Definition. 1. Model-based FTFC system:

A model-based FTFC system is a model-based flight control system which can obtain

failure knowledge using a fault detection & isolation unit, and improve the situation awareness of a pilot or an automatic controller by specifying the current safe flight envelope. It can accomplish the demanded flight tasks after accommodating ongoing fault scenarios using a non-conventional reconfigurable flight control law. □

By definition, a model-based FTFC system contains an aerodynamic model. This model is essential when designing and applying a model-based FTFC system. An accurate aerodynamic model is required for at least one of the following components: FDI, reconfigurable control laws, and flight envelope protection, see Figure 1.3. In this thesis, the work is limited to investigating the design of a model-based FTFC system, to which an accurate aircraft aerodynamic model is crucial.

1.2.1. MODEL REQUIREMENTS FOR MODEL-BASED ADAPTIVE FLIGHT CONTROL

The existing control algorithms that can achieve reconfiguration within an FTFC system can be classified into five categories according to the different reconfiguration mechanisms that they use: optimization, switching, matching, model following, and adaptive compensation [175]. Gain scheduling (GS) of linear control, linear parameter varying (LPV) model based control and sliding mode control (SMC) fall into the category of switching based methods, which accomplish reconfiguration by switching. To design a controller using LPV models, a nonlinear system model is unnecessary. Instead, one or more linear state space models are required to design a controller using a LPV model based method. Similarly, in the implementation of most SMC approaches, only a linear baseline model is required to determine the sliding surface. The reconfigurable control approaches based on adaptive compensation mainly include direct and indirect adaptive control methods. In an indirect adaptive control approach, a model-based adaptive control algorithm requires an accurate online aerodynamic model, which is assumed to be able to capture any changes in the flight performance caused by maneuvers, structural failures or other aircraft damage. In this thesis, we focus on model-based adaptive flight control algorithms for indirect adaptive control because a model-based control algorithm has a number of advantages.

The model-based adaptive flight controller can have a higher control performance than a model-free flight controller because the former can be designed to require relatively low control power or effort relying on knowledge of the aerodynamic model [162][p.37]. In model-free flight control approaches, for example in incremental control approaches and robust nonlinear control methods, all dynamics of the controlled system, excluding actuator dynamics, are inverted or compensated to make the closed-loop system an ideal reference tracking system, i.e., an identity system. However, some nonlinearities or nonlinear dynamics in the open-loop aircraft system are useful feedback terms which can contribute to the stability of the system. For instance, damping terms may be present in the aerodynamic model, and these potentially useful nonlinearites should not be canceled in the controller design [162][p.37]. In a model-based adaptive flight control design, e.g., model-based adaptive backstepping, the stable dynamic terms mentioned above can be maintained in the final closed-loop system without being canceled by the controllers. This results in a high performance controller

in the sense that it requires relatively low control effort [162][p.37].

An accurate aerodynamic model plays an essential role in determining the performance of a model-based adaptive flight controller. To allow the controlled system to adapt for model uncertainties or other model changes, the aerodynamic model, which is the heart of the control system, is required to be updated in real-time [27, 47, 87, 163]. Due to the limited computational capabilities of the onboard computers, it is essential to choose or develop a recursive system identification algorithm requiring low computational cost [87].

An online aerodynamic model derived using a certain identification method can provide fault information on structural failures [181]. Given the estimated aircraft states, the aircraft model identification problem is reduced into a function approximation problem when using some type of identification methods [88]. Model structure selection is one of the main issues when using a function approximator. Many aerodynamic model identification methods, like the polynomial basis based method [87] and the neural partial differential method [29], can yield a model with clear physical interpretation due to the model structures that they use. Using these methods, the control derivatives of all control surfaces and the stability derivatives can be either directly known from the identified model or can be indirectly calculated from the model, and the structural failure information can be attained by observing and analysing the changes in the control derivatives.

To achieve full-envelope modular adaptive control, a global aerodynamic model with local updating capability is indispensable [163]. In the implementation of such a full-envelope control approach, a nominal full-envelope model is usually trained in advance using flight test data and, possibly, wind tunnel data. Subsequently, a local part of the global model is updated using local measured data to account for any possible uncertainties or other model changes in real flight.

1.2.2. MODEL REQUIREMENTS FOR FLIGHT ENVELOPE PROTECTION

To achieve FTFC, not only a reconfiguring controller is needed, but also a flight envelope protection unit. The latter prevents the aircraft from entering an upset flight condition. Flight envelope protection can be categorized into two branches: static and dynamic [85, 164]. Dynamic flight envelope protection can also be called online flight envelope protection OFEP.

Some methods for static flight envelope protection (SFEP) need an aerodynamic model of the aircraft that is valid in the entire flight envelope, and nearly all of them need a function approximation algorithm. A SFEP is based on an *a priori* knowledge of the aerodynamic model and thus has no real-time adapting capability. It is usually incapable of accounting for severe model changes occurring in a real flight. In the first step of static envelope estimation, attainable equilibrium sets or achievable trim points need to be obtained using approaches such as wind tunnel testing, real flight test experiments and high-fidelity model-based computation [19, 52, 66, 149]. According to the literature, the model-based computation method relies on an accurate aerodynamic model which is valid in the entire flight envelope. In the second step, the attainable equilibrium sets obtained in the previous step are approximated and expanded to build a library for FEP in the *entire* flight envelope. Central to the problem in step two is a model approximation

problem using the available data. Potential function approximation approaches that might be suited for this purpose include a polynomial based method, multivariate B-splines, fuzzy logic, neural networks and other kernel methods.

The methods for OFEP also require an accurate aerodynamic model which is valid within the entire flight envelope. The preferred methods for OFEP include formulating the flight envelope estimation problem into a reachability problem [3, 72, 85, 92, 127, 150, 164]. A reachability analysis, i.e., solving the Hamilton-Jacobi-Bellman (HJB) equations which are associated with a time-dependent integral cost function in real-time, is conducted based on an accurate aerodynamic model. No matter what following solvers, e.g., level set methods and cost function approximation methods, are chosen to perform the reachability analysis, the aerodynamic model applied here has to be accurate and valid for the entire flight envelope. Furthermore, to take into account aircraft failures such as structural damage, the aerodynamic model has to be an online, globally valid model which can reflect sudden model changes caused by possible ongoing failures.

Note: the implementation of FEP is out of the scope of this thesis and only the global model identification methods which can be used for FEP are studied. The candidate methods are modified or improved in Part I of this thesis to satisfy the requirements from FEP, i.e., high approximation power and high efficiency.

1.2.3. RECURSIVE AERODYNAMIC MODEL IDENTIFICATION

A powerful aerodynamic model identification method is indispensable for both model-based adaptive flight control and FEP. For the purpose of full-envelope modular adaptive control and FEP, a global model identification approach is needed, and for adaptive model-based control and OFEP, the model identification algorithm needs to be computationally efficient to allow for real-time model updating. Some state-of-the-art model identification approaches are briefly reviewed in this section.

Many different methods have been proposed in the literature for aircraft model identification. A joint method for state and parameter estimation is also called the one-step method, as here the states and aerodynamic parameters are estimated simultaneously in one procedure [110]. There are many different one-step model identification routines. An example is the maximum likelihood identification routine, which solves the joint estimation problem by solving a global nonlinear optimization problem, i.e., maximizing a likelihood function composed of output or prediction errors [93]. Due to the computational demanding property caused by global nonlinear optimization, only a very few joint identification approaches are applicable online. One example of the joint state and parameter estimation algorithms that can be implemented in real-time is a nonlinear filtering method developed at the German Aerospace Research Center DLR [51, 61]. Though online implementable, this model estimation algorithm is still computationally very demanding, especially when a large number of aerodynamic parameters is involved.

An alternative to a joint identification method is to use a two-step method. In a two-step method, the states and the aerodynamic parameters are estimated in two consecutive steps [108]. The first step is also a joint state and parameter estimation problem, however, this problem is relatively easy to solve since the aerodynamic parameters, the

number of which is usually large, are not involved [86][p.116]. The estimation accuracy of the first step is easy to guarantee, this is because only kinematic models of aircraft are used in the first step, and the complex yet uncertain aerodynamic model is not included. In the second step, the aerodynamic model identification problem is reduced into an *equation-error* parameter estimation problem [68][p.216] once the aircraft state has been estimated in the first step [106–108]. The aerodynamic model structure can be assumed *linear-in-the-parameters* without sacrificing the reliability of the model in most applications [106, 107, 109, 110]. Many regression algorithms can be applied to solve the *equation-error* parameter estimation problem in the second step. Due to the relative simplicity of the regression algorithms, two-step methods result in a lower computational load and are better suited for being implemented online.

Aerodynamic model identification can also be performed in the frequency domain, but not all the frequency-domain identification methods are suitable for real-time use [68, 154]. Examples of methods which are suited for real-time implementation can be found in [102, 104, 105, 134]. However, in this thesis we focus on identification methods in the time domain. More specifically, the research scope of this thesis is limited to developing powerful advanced algorithms for the *equation-error* parameter estimation problem in the second step of the two-step routine.

Among all the system identification methods, such as function approximation approaches, there are many potential candidates that are powerful and thus might be suited to providing accurate global aerodynamic models for model-based adaptive flight control and OFEP. These methods can be divided into parametric and nonparametric methods. The nonparametric methods include a number of reinforcement learning methods, e.g., many neural networks, most of the kernel methods, support vector machines, and fuzzy logic modeling methods [28, 68, 133, 166]. The parametric methods include a few kernel methods and functional approximation methods such as polynomial basis-based methods and variant spline polynomial basis-based methods.

One example of the parametric function approximation methods, which may be capable of providing a real-time aerodynamic model for the purpose of model-based adaptive flight control and OFEP, is the multivariate simplex B-splines (MVSB) method. The MVSB method can be used to identify a global aerodynamic model of an aircraft. Using this method, the fault detection and isolation (FDI) unit should be able to tell which components or control surfaces are currently malfunctioning by judging the identified parameters which represent the effectiveness of each control surface. Unlike tensor product splines, the MVSB method is able to use scattered datasets [73]. The approximation power of this method depends on the density of the simplices, i.e., subdomains, and the polynomial order within each simplex. In this method, the B-spline bases, i.e., Bernstein basis polynomials, have been proven to be stable due to the fact that the barycentric coordinates of the evaluation data points are defined based on a local simplex within a triangulation [73]. The MVSB method leads to predictable model outputs, i.e., their regions are predictable, once the B-coefficients are given [10, 73].

The computational efficiency of the state-of-the-art MVSB methods still needs to be enhanced before it can be applied in a real application where an accurate real-time model is required. Though the batch type MVSB method has been applied to modeling a set of scattered datasets for F-16 aircraft [35], this method has not yet been widely

investigated in the field of aircraft modeling. Although a recursive identification method called equality-constrained recursive least squares (ECRLS) has been proposed for the MVSB methods [37], all existing MVSB methods still have a high computational load if the selected spline model is of large scale. The computational efficiency of the MVSB method therefore needs to be improved before it can be used in real-time to provide an online aerodynamic model for the purpose of fault detection, i.e., monitoring the effectiveness of a control effector, and to provide a model for updating the safe-flight-envelope model or updating of the controller parameters constituting the reconfigurable control laws.

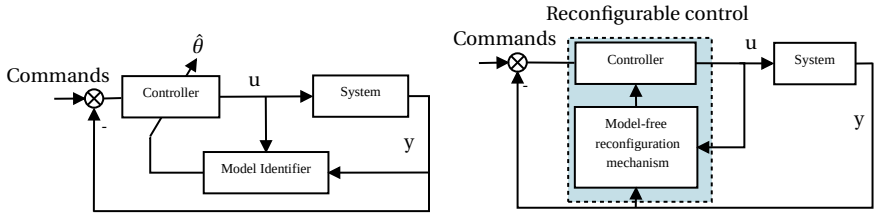
A recursive kernel method provides an example of a nonparametric method. This method could also be used as a powerful model identification tool to detect failures or as a foundation for a global prediction model that is used to determine a safe-flight-envelope for an aircraft. There is a well-known open issue associated with the regular recursive kernel methods: how to determine the number, the centers and radius of the kernels [28]. The optimality of the kernels has a significant influence on the approximation power of the regular recursive kernel methods. In the existing literature, researchers usually use the $k-means$ clustering method to determine the centers of the kernels [28, 126]. However, this method cannot guarantee the optimality of the kernel, and will probably limit the approximation power of the recursive kernel methods. Therefore, a better way to determine the kernel related parameters needs to be found.

1.2.4. RECONFIGURABLE FLIGHT CONTROL

A generic FTFC system should contain a reconfigurable control unit, which should be able to extract all the remaining control authorities of an aircraft and enable a closed-loop system to respond appropriately to the reference commands regulated by the FEP unit, see Figure 1.3. As mentioned in Section 1.2.1, there are many different reconfigurable flight control laws, the reconfiguration mechanisms of which that help to accommodate sudden changes in the aerodynamic model caused by structural or actuator faults include robust switching and adaptive compensation [175]. According to the requirements for an aerodynamic model, the reconfiguring flight control laws can be classified into two categories: model-based control laws and model-free control laws, see Figure 1.4. In Figure 1.4(a), a model identifier is designed to provide a real-time accurate aerodynamic model for the adaptive controller. The reconfiguration of this type of controllers is ascribed to the compensation of the real-time model, which accurately captures the changes in the behavior of the flight performance. The reconfigurable controller shown in Figure 1.4(b) does not require an aerodynamic model, but an extra reconfiguration mechanism such as a switching or matching logic is needed [162, 175].

Both model-based and model-free reconfigurable control laws have advantages and drawbacks. Model-free control laws include the sliding mode control laws, linear parameter varying methods and conventional PID control laws. The drawback of using these kinds of control laws is that controller switching or gain-scheduling is always needed to achieve the controller reconfiguration. Model-based control laws include adaptive nonlinear control laws such as adaptive nonlinear dynamic inversion (NDI) and modular adaptive backstepping. Model-based nonlinear control approaches have a few drawbacks. Firstly, they are sensitive to model inaccuracies, however, even the

most powerful model identification schemes might yield a discrepancy between the identified onboard aerodynamic model and the true system during acute failures or actions requiring high maneuverability [2, 68]. Although a few robust nonlinear control strategies can be incorporated in the model-based controller [113, 162] to get better tracking performance, the robust control laws are susceptible to an assumption that the model discrepancy is within a chosen boundary. Secondly, an online aerodynamic model usually results in a high computational load, depending on the complexity of the chosen system identification methods, which makes it necessary to make a trade off between model accuracy and computational load when choosing a model structure and the identification algorithm [68, 102, 163]. Thirdly, the stability of the closed-loop system cannot be guaranteed by model-based adaptive control due to the weakness of the *certainty equivalence* condition [162][p.70]. The convergence of model parameters is hard to guarantee when identifying a closed-loop system without persistent excitation. Finally, model-based nonlinear control approaches are more difficult to certify than model-free flight control methods [45].



(a) Using indirect/model-based adaptive control algorithm.

(b) Using reconfigurable control method which requires no online model.

Figure 1.4: Control structure comparison between model-based and model-free reconfigurable control methods.

Both model-based and model-free reconfigurable flight control laws were investigated for the research presented in this thesis. With regard to model-based control, a new model-based adaptive flight control system was designed for an F-16 aircraft by combining nonlinear dynamic inversion with online aerodynamic model identification using MVSB, see [156]. This work is not included in this thesis, however, and the control related part of this thesis mainly focuses on model-free reconfigurable flight control. More specifically, acceleration measurements based incremental flight control laws form the main focus. We also investigated how the estimation accuracy of the control effectiveness matrix affects the performance of an adaptive incremental flight controller.

Recently, the incremental type of control laws, for example the incremental NDI [129] and incremental backstepping [44], have attracted a large amount of attention in the domain of aircraft flight controller design. Advantages of incremental control laws are that they are not subject to model mismatches due to the nature of the control approximation. Nevertheless, when used in a real flight control design, the control performance of such incremental control approaches may still be affected by

the identification accuracy of the control derivatives, especially when the aircraft model has large parameter variations in the entire flight envelope or the sampling rate of the flight control computer is low.

More recently, the adaptive NDI control law and the sliding mode control law have been applied in a reconfigurable control for vehicle emergency relief (RECOVER) model to improve the performance of a FTFC system in aircraft [6, 87]. An alternative to this is the sensor-based backstepping (SBB) control approach, which is a special type of incremental and high-gain control algorithm. The SBB method, because of its incremental nature, is assumed to be able to tolerate large model uncertainties caused by sudden damage occurring in an aircraft. Compared to the model-based methods, the advantage of the SBB control algorithm is that it does not require accurate online aerodynamic model information, the reliability of which cannot always be guaranteed during a transition period when major structural or actuator failures happen to an aircraft or when high maneuverable flight tasks are performed. Instead, the SBB control method uses the derivatives of the controlled variables, which can be measured using sensors such as angular accelerometers. Unlike other nonlinear incremental methods, the SBB method has been developed based on singular perturbation theory and Tikhonov's theorem [46, 59]. As a consequence, the SBB control algorithm has a time-scale tuning parameter due to the approximation controller solution which uses the singular perturbation theory, and this parameter can therefore be used to help simplify the tuning process of the controller parameters.

The SBB approach is investigated in Falkena et al.[46] and applied to design a flight controller for small aircraft such as the Diamond DA 42. However, how this control approach can be synthesized with control allocation techniques, with which a control effectiveness matrix is associated, still needs to be investigated before it can be used to design a flight controller for a large civil aircraft such as the Boeing 747-200, which has many redundant control surfaces. In addition, the influence of the sensor noise, which is associated with the sensor systems of the Boeing 747-200 aircraft, on the flight control performance needs to be investigated further. Finally, the robustness of such an SBB control approach to sudden structural or actuator failures, for example, rudder runaway and engine separation failure, has yet to be tested, and needs to be studied in depth before being applied in practice.

1.3. MAIN RESEARCH QUESTION

The main research question of this thesis is:

> **How can an advanced fault-tolerant flight control system be designed to increase the survivability of an aircraft?**

Following the literature, a generic model-based FTFC system requires a powerful aerodynamic model identification approach and should contain a reconfigurable flight control component. Therefore, the main question was split into two subquestions:

1. **How can the candidate function approximation methods, i.e., MVSB and kernel methods, be improved in terms of their approximation accuracy and computational efficiency, to meet the needs of model-based adaptive control and OFEP?**

2. **What are the benefits of using an acceleration measurements-based control approach, i.e., the sensor-based backstepping, as an alternative to a model-based adaptive control approach, when designing a reconfigurable flight controller to deal with aircraft failures in a generic FTFC system?**

The two subquestions formulated above are closely related. The first is associated with model-based adaptive flight control and FEP, and the second contributes to a reconfigurable flight control unit. As mentioned in Section 1.2, a reconfigurable flight control unit, which is designed using either a model-based adaptive control approach or an incremental control scheme, and an FEP unit are two of the most important components in a generic FTFC system. An FEP unit is in charge of providing the pilot or the automatic controller with the remaining control authority of a post-failure aircraft by showing them the reachable reference commands after evaluating and analyzing the ongoing structural or actuator failures. In addition, the sensor-based backstepping (SBB) control approach, which is the kernel of the internal controller, should guarantee that the outputs of the aircraft are tracking the reference commands provided by the FEP unit.

As mentioned in Sections 1.2.1 and 1.2.2, model-based adaptive flight control approaches have many advantages, and both a full-envelope modular adaptive control and an FEP require a powerful global model identification scheme. Therefore in this thesis, global model identification methods were investigated for the aforementioned purposes with a focus on improving approximation accuracy and computational efficiency.

1.4. SCOPE AND LIMITATIONS

This thesis contains two parts: global aerodynamic model identification part and acceleration measurement-based incremental nonlinear control part. Part One focuses on presenting recursive global aerodynamic model identification methods developed for model-based adaptive reconfigurable control and model-based flight envelope protection. Part Two focuses on exploring the benefits of using acceleration measurement-based incremental nonlinear control laws, which do not rely on complete, accurate aerodynamic model knowledge, for fault-tolerant control purposes.

The scope of Part One of this thesis is limited to the study of two types of global model identification methods: 1) the recursive identification method using multivariate simplex B-splines; and 2) the recursive, adaptive, kernel method inspired by the support vector regression method. The model identification methods studied and developed are intended to provide accurate offline or online global aerodynamic models for model-based adaptive flight control approaches and static or online FEP. The two types of global model identification methods developed, i.e., the improved recursive MVSB methods and adaptive kernel methods, have yet to be incorporated into a flight control system or into a FEP problem. In this thesis, these two model identification methods are

only validated independently. Investigating the reliability of the model under model-based control or a FEP framework was considered beyond the scope of the thesis. Furthermore, it was assumed that the global model identification methods developed in this thesis contained two training steps. Firstly, an offline nominal global aerodynamic model was trained using flight test data, wind tunnel data or data from a high fidelity model. Secondly, the aerodynamic model was updated locally using current attained flight test data.

In Part Two of this thesis, two incremental flight control approaches are investigated and extended to deal with fault scenarios. In addition, a model identification method using MVSB is applied to identify the control effectiveness matrix which is comprised of control derivatives. How the control effectiveness matrix influenced the overall control performance of the incremental controller was also investigated.

1.5. CONTRIBUTIONS

Three main contributions to FTFC are made with the research presented in this thesis. Firstly, three recursive global model identification methods are developed. A substitution based strategy and a recursive sequential strategy are proposed to speed up the B-spline basis based function approximation methods. In addition, a new type of recursive kernel method is proposed for use after synthesizing the support vector selection techniques into the regular kernel methods. These model identification methods can be used either for the purpose of OFEP or to provide an accurate global aerodynamic model for the design of model-based reconfigurable flight control systems.

Secondly, an acceleration measurements-based control approach called sensor-based backstepping, which is a state-of-the-art control approach to a complex nonlinear system and was developed over the past decade, see Hovakimyan et al.[59] and Falkena et al.[45], is extended in its application. The control approach is applied to the RECOVER model with a focus on dealing with sudden model changes caused by structural or actuator failures. To accomplish this application, many related and specific techniques are investigated, e.g., control allocation, sensor noise rejection technique, determination of the overall control structure, i.e., hierarchy.

Thirdly, a tensor-product multivariate simplex B-spline method is extended in this thesis and applied in an incremental flight control framework to provide an estimation of the control effectiveness matrix for an incremental controller. The influence of the control effectiveness matrix on the overall flight control performance is investigated based on a nonlinear F-16 high fidelity model.

The list of journal papers, proceeding papers, and book chapters are given in Appendix H.4.

1.6. THESIS OUTLINE

A thesis outline is shown schematically in Figure 2.1. The body of the thesis is divided into two parts. In the first part, three recursive global model identification approaches are presented in Chapters 3, 4 and 5. The second part, Chapters 6-9, focuses on reconfigurable control approaches. Conclusions and final recommendations are presented in Chapter 10.

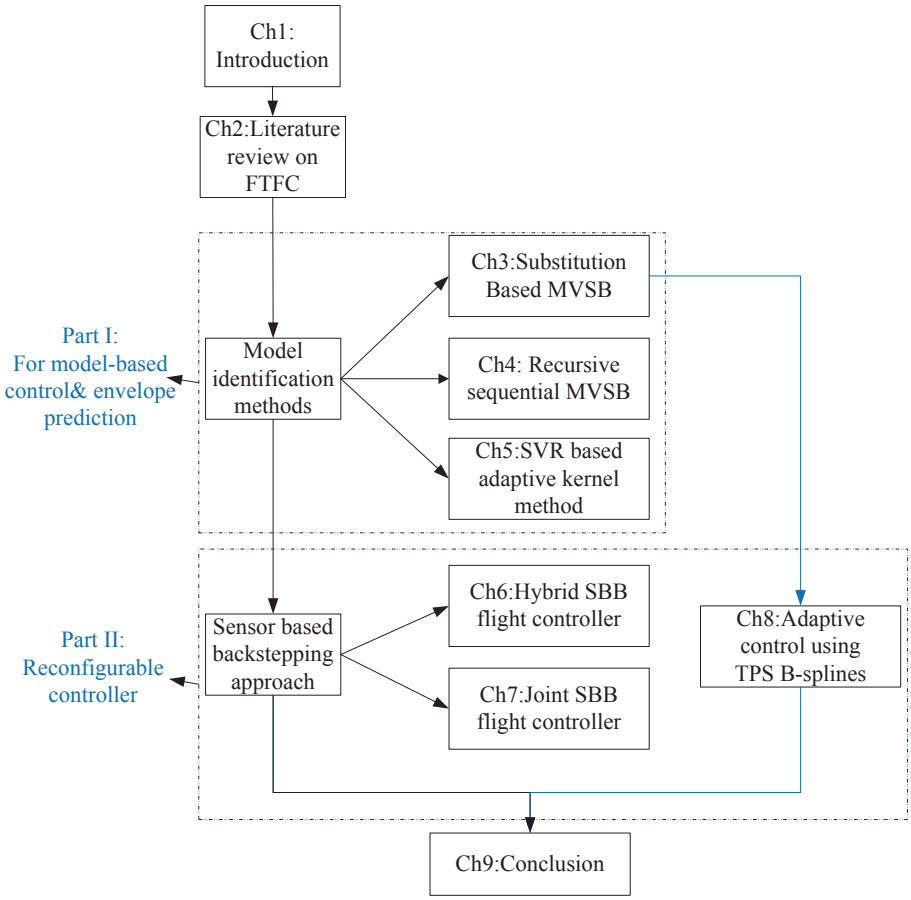


Figure 1.5: Diagram for the contents of each chapters.

The existing baseline methods applied in the field of online aerodynamic model identification, nonlinear reconfigurable control, and baseline methods applied in other components included by a fault-tolerant flight control FTFC system are discussed in **Chapter 2**.

A substitution based recursive identification method using multivariate simplex B-splines (MVSB) is developed in **Chapter 3**. In this method, the equality constraints associated with the smoothness property of simplex B-splines are converted to make the whole recursive identification method free-of-constraint. **(Paper [138, 156])**

A new recursive sequential strategy to reduce the computational cost required for the multivariate simplex B-splines based recursive identification method is presented in **Chapter 4**. In this method, the calculation of a global covariance matrix is avoided by limiting the model updates at each time instant into local simplex updates. In addition, a

strategy to impose the smoothness constraints on the triangulation is proposed. (**Paper [141]**)

A support vector regression based adaptive kernel method for realizing recursive global aerodynamic model identification is proposed in **Chapter 5**. In this method, a state-of-the-art recursive reduced support vector regression algorithm is applied to a regular kernel method to help determine an optimal set of kernels. In addition, the extension of the local kernels is investigated with the aim of enhancing the approximation power locally. (**Paper [144]**)

The design of a hybrid sensor-based backstepping (SBB) angular controller is presented in **Chapter 6**. The sensor-based backstepping control approach is employed in the body angular rate loop in combination with a simplified control allocation strategy. The regular nonlinear dynamic inversion control method is used to design the controller for the angular loop. In addition, the flight path controller is designed using regular PID to complete the design of an autopilot for a Boeing 747-200. (**Paper [139]**)

The design for a joint SBB angular controller is presented in **Chapter 7**. The SBB control approach is employed to design a controller for the body angular rate loop. Subsequently, the body angular rate controller is augmented using the backstepping strategy to design an outer-layer controller for the angular loop. This controller is validated using the RECOVER model, which consists of a set of benchmarks containing aircraft fault scenarios. (**Paper [140, 143]**)

A new tensor-product multivariate simplex B-spline method is extended from a single-dimension case to a multi-dimension case in **Chapter 8**, and this method is compared with standard simplex B-splines with a focus on the computational efficiency and approximation power. In addition, the tensor-product MVSB model identification method is incorporated into an adaptive incremental nonlinear flight controller, where the simplex B-spline model provides the controller with a control effectiveness matrix. The flight performance of the overall closed-loop flight control system is evaluated by simulations using a high fidelity F-16 nonlinear model. (**Paper [142]**)

The conclusions of this work and recommendations for future work are given in **Chapter 9**.

2

Literature Review on Fault-tolerant Flight Control Components

A literature review on the methods used to design the basic components of a generic FTFC system is presented in this chapter. An overview of a generic FTFC system is given in Section 2.1. A classical online model identification routine called two-step method [108] is introduced in Section 2.2 with an introduction to state estimation using extended Kalman filters. Real-time aerodynamic model identification issues are revisited in Section 2.3. The state-of-the-art methods in the field of fault detection and diagnosis (FDD) are reviewed in Section 2.4. Reconfigurable flight control approaches are revisited in Section 2.5. Finally, flight envelope protection and prediction issues are discussed in Section 2.6.

2.1. Basic Components in Model-based Fault-Tolerant Flight Control

The next generation of aircraft requires higher levels of survivability. Military fighter aircraft are devised to have high agility for air combat purposes, which makes the airframe structure or the actuators more complex. As a consequence, a higher probability exists that one component of the complicated system can go wrong when the aircraft is suffering from an emergent threat. Regular tasks such as launching a missile can also cause a large instant disturbance to the aircraft configuration. Compared to fighter aircraft, safety plays a much more important role in the design process of large civil aircraft because the safety of large numbers of passengers relies on the safety of such civil aircraft. Structural failures of the airframe, actuator failures, engine failures and sensor faults need to be addressed using advanced fault detection or accommodation techniques.

To be more resilient, current fly-by-wire control systems of modern aircraft, e.g. Boeing 787, Airbus 380, F-35 and Eurofighter, have been devised to have two new components: 1) a sensor or actuator fault detection & diagnosis unit; 2) a flight envelope protection unit. An excellent and extensive literature review on fault-tolerant flight control system can be found in [175], which shows that the tendency to improve aircraft safety is drawing increasing attention from the aerospace research community.

Two main types of FTFC systems exist: model-based FTFC systems and FTFC systems requiring no system model. However, compared to a model-free FTFC system, the model-based FTFC system has been shown to have unique capabilities and advantages both from the aspect of reconfigurable flight control [41, 87, 99, 174] and from the aspect of real-time aerodynamic model-based online flight envelope protection [54, 85, 127, 164]. Therefore, only the model-based type of FTFC system is discussed in this chapter.

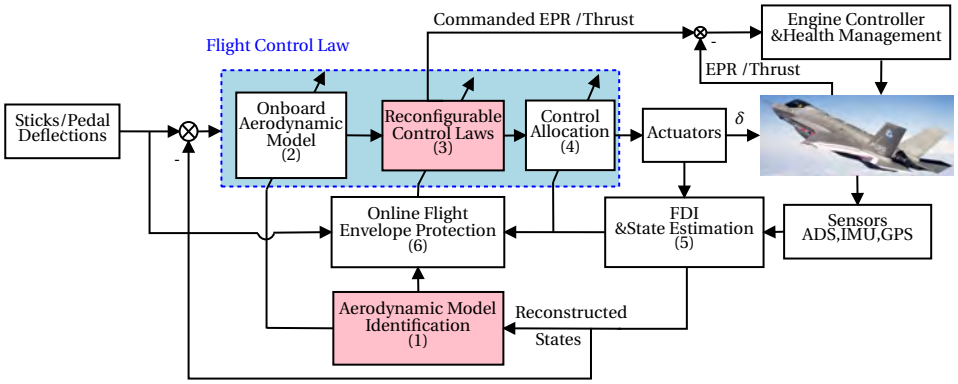


Figure 2.1: Block diagram of a conceptual modern model-based flight control system.

An overview of a model-based fault-tolerant flight control system, a definition of which was given in Section 1.2, is shown in Figure 2.1. As indicated in Figure 2.1, an FTFC system using model-based control laws usually consists of four basic components and two additional advanced units. The four basic components are: 1) an aerodynamic model identification (AMI) unit which may include state estimation; 2) an adaptive model reconstruction unit; 3) a reconfigurable control law unit and 4) a control allocation unit. The two additional units are: a fault detection & isolation unit and a flight envelope protection unit. The latter two units are usually introduced in the modern FTFC systems to enhance the survivability of an aircraft further. A fault detection & isolation unit is usually closely correlated to flight path reconstruction. Specifically, the fault detection & isolation unit can provide failure information, e.g., failure type and a rough estimation of the damage level, to the flight envelope protection unit and the control allocation unit. The online flight envelope protection unit can predict the current safe flight envelope in case of failures, and adjust the reference commands fed to the reconfigurable control laws to make them consistent with the reduced safe flight

envelope.

Among all of the six components, see Figure 2.1, the AMI part which sometimes also involves state estimation and the reconfigurable control laws part are the main focus of this dissertation, and will be discussed in detail. The other three blocks shown in Figure 2.1 are also briefly introduced in this chapter for the sake of completeness.

2.2. STATE ESTIMATION, JOINT METHOD AND TWO-STEP METHOD

2.2.1. JOINT AND TWO-STEP AERODYNAMIC MODEL IDENTIFICATION

There are many different aircraft model identification algorithms ranging from time-domain methods to frequency-domain methods. As mentioned in Section 1.2.3, the focus of this thesis is the identification methods in the time domain. In the time domain, the aircraft system identification methods can be classified into one-step (joint) methods and two-step methods, depending on whether the system states and the unknown aerodynamic parameters are dealt with in one single identification process or two.

The joint approach treats the unfiltered aircraft states and unknown aerodynamic parameters equally in one single process. One example of this is the maximum likelihood method [68]. An optimization solver such as a gradient-based optimizer is usually required to solve a global optimization problem when using this type of methods. The joint model identification algorithms have many merits and have been widely applied in many fields [68]. This type of model identification methods is able to deal with a system with process noise and the output fitting errors can be assumed correlated with the model parameters. However, this type of identification methods result in a high computational load if the number of unknown parameters is large due to the computational complexity caused by global optimization. The number of unknown parameters in a physical model of a fixed-wing aircraft varies from tens to one hundred, see [68, 88], therefore, it is not a wise choice to use a joint model identification method in this case. Due to the computational complexity introduced by the required global optimization task, there are quite few joint aircraft model identification methods that can be applied in real time. A representative example of such a method is a nonlinear filtering method based on Kalman filters or unscented Kalman filters [51, 61].

An alternative to the joint method is the two-step method, which separates the state estimation from the identification of the unknown aerodynamic parameters [108]. This two-step method is developed based on the fact that the aircraft states have clear physical interpretation and they are able to be properly estimated using only the kinematic equations with the aid of the Kalman filtering. The two-step model identification method presented in [108] is capable of providing a real-time aerodynamic model to a model-based 'Reconfigurable Control Laws' unit and an 'Online Flight Envelope Protection' unit, see Figure 2.1.

2.2.2. FLIGHT PATH RECONSTRUCTION USING KALMAN FILTERS

Flight paths are required for many subsystems in a model-based FTFC system, and need to be reconstructed from aircraft states. The aircraft states can be estimated from the measured inputs, i.e. specific forces and angular accelerations, and measured outputs. The preferred state estimation algorithms include an extended Kalman filter

(EKF) and an iterated EKF. This dissertation does not focus on the algorithms, and the reader is referred to [86] for more details. However, to be ready to implement the EKF, the kinematic equations of a fixed-wing aircraft and the correspondingly augmented expressions are given in Appendix A. In addition, how the aircraft states can be estimated when an aircraft encounters a time-varying wind, i.e., a turbulence, was investigated and was reported in Appendix B. A generalized version of the kinematic equations were developed to enable an EKF to deal with time-varying wind speeds. The newly formulated kinematic equations are generic because the assumption that the wind speed is time invariant is removed.

It also should be noted that the fly-by-wire control system of a modern aircraft normally no longer needs an independent state estimation component because most of the measurement units currently used are so advanced that they can provide state estimations automatically. Additionally, many FDD methods are closely related to Kalman filtering and state estimation, see Figure 2.1, thus fault detection and reconstruction of the correct states can be achieved simultaneously.

2.3. REAL-TIME AERODYNAMIC MODEL IDENTIFICATION

An accurate onboard aerodynamic model is required for both the model-based fault tolerant flight control problem and dynamic prediction problem of the safe flight envelope. The field of recursive identification can be split into parametric methods and nonparametric methods, in this thesis the focus is placed on the parametric methods. One promising example of such online parametric identification methods is the ordinary polynomial basis based (OPBB) method. This method has been intensively studied by Klein, Morelli et al. [69] and other researchers, see [68, 88, 100, 137].

Other recursive parametric model identification methods include simplex B-splines, radial basis function neural networks (RBF-NN), and other kernel based methods, which will be introduced in Chapters 3, 4 and 5 rather than in this chapter.

2.3.1. RECONSTRUCTION OF DIMENSIONLESS FORCES AND MOMENTS

Dimensionless forces:

As preparation for the aerodynamic model identification process, the non-dimensional moment coefficients can be reconstructed using the measured specific forces and states as follows:

$$C_X = \frac{X}{\frac{1}{2}\rho V^2 S} = \frac{mA_{x\text{AMI}}}{\frac{1}{2}\rho V^2 S} \tag{2.1a}$$

$$C_Y = \frac{Y}{\frac{1}{2}\rho V^2 S} = \frac{mA_{y\text{AMI}}}{\frac{1}{2}\rho V^2 S} \tag{2.1b}$$

$$C_Z = \frac{Z}{\frac{1}{2}\rho V^2 S} = \frac{mA_{z\text{AMI}}}{\frac{1}{2}\rho V^2 S} \tag{2.1c}$$

Dimensionless moments:

The non-dimensional moment coefficients are reconstructed using the measured states as follows:

$$C_{\mathrm{l}} = \frac{L}{\frac{1}{2}\rho V^2 Sb} = \frac{\dot{p} I_{xx} + qr\left(I_{zz} - I_{yy}\right) - \left(pq + \dot{r}\right) I_{xz}}{\frac{1}{2}\rho V^2 Sb} \tag{2.2a}$$

$$C_{\mathrm{m}} = \frac{M}{\frac{1}{2}\rho V^2 S\bar{c}} = \frac{\dot{q} I_{yy} + rp\left(I_{xx} - I_{zz}\right) - \left(p^2 - r^2\right) I_{xz}}{\frac{1}{2}\rho V^2 S\bar{c}} \tag{2.2b}$$

$$C_{\mathrm{n}} = \frac{N}{\frac{1}{2}\rho V^2 Sb} = \frac{\dot{r} I_{zz} + pq\left(I_{yy} - I_{xx}\right) - \left(qr - \dot{p}\right) I_{xz}}{\frac{1}{2}\rho V^2 Sb} \tag{2.2c}$$

where $\dot{p}$, $\dot{q}$ and $\dot{r}$ are angular accelerations. In the work presented in this thesis, the angular accelerations are derived from the rotational rates, but they will be measured using angular accelerometers in future realistic applications, e.g., acceleration measurements-based incremental flight control. For the meaning of the other symbols used in the above equations, the reader is referred to the list of symbols.

2.3.2. AERODYNAMIC MODEL STRUCTURE SELECTION

For fault-tolerant flight control purposes, an aircraft aerodynamic model is required to be updated in real-time. Model identification commonly includes two parts: model structure selection and parameter estimation. The research reported in this thesis is mainly focused on parameter estimation. To allow for real-time identification of an aerodynamic model, parametric methods such as polynomials, neural networks, splines and most of other recursive kernel methods are required to determine the model structure in advance. In addition, it is necessary to make a trade off between modeling accuracy and computational complexity when selecting a predefined model structure.

In the ordinary polynomial basis based (OPBB) identification method, the model structure selection problem can be solved by ranking the polynomial terms according to their effect factors using the orthogonal least squares methods [68, 137]. The basic idea is to choose a subset of regressors from a pre-determined regressor pool based on the feedback from the output fitting errors. A specific pre-defined aerodynamic model structure for a Boeing 747 aircraft when using polynomial-based model identification method [88, 89] is given in Appendix C to illustrate this method.

The model structure selection task for simplex B-spline basis based model identification methods is reduced into determining the optimal triangulation, function dimensions and the per-simplex polynomial orders [33, 35]. For recursive kernel methods, model structure selection becomes determining the number of kernels, and the bandwidth and center positions of the kernels [28].

2.4. FAULT DETECTION AND DIAGNOSIS

The FDD unit is very important in a modern fault-tolerant flight control system [181–183]. It should have the following functions. Firstly, it should be able to tell the controller which sensor or actuator goes wrong. Secondly, it is required to tell the controller which

failure the aircraft is currently suffering from, which ensures that the flight control mode is switched to the correct mode. This information should also be communicated to a control allocation unit to make sure that only the effectors which are still available are involved in the new control distribution. Thirdly, under any sensor or actuator failures, the FDD unit can be required to provide analytical redundancy and provide true estimates of the sensor or actuator measurements to the flight path reconstruction unit.

In [94], Marzat et al. give a very complete overview of fault detection in aerospace systems. In the field of practical application of fault diagnosis, modern FDD techniques have been used for the flying A380 aircraft, which currently is the only practical application of the model-based fault detection method for used in commercial aircraft. The actual implementation of the FDD method is described in [53]. More recently, Lavigne et al.[75] have performed more validation tests on model-based fault detection approaches using A380 flight data. In addition to the above, there are many academic methods that can be used to detect aircraft sensor or actuator failures, for example, robust observers, sliding mode observers, decoupling strategies, and geometric approaches.

The methods for FDD can be categorized into three branches: air data fault detection; inertial measurement unit (IMU) data fault detection and simultaneous fault detection for both the air data sensor and the IMU. The existing approaches all need measurements from both the air data sensor (ADS) and the IMU. One promising branch of the available FDD methods is a physical model based approach [86, 159–161], which exploits the properties of a physical aircraft model such that a transparent physical interpretation exists.

Among the physical model based FDD methods, the preferred methods used for the research reported in this thesis are those developed by Van Eykeren et al. [159, 161], where the kinematic relations of an aircraft are used in the adaptive form of an extended Kalman filter. By observing a metric derived from the innovations of the EKF, the approach in [159] is able to monitor the performance of each redundant sensor. Subsequently, the failed sensors can be isolated according to this metric. The work in [159] forms part of the contribution to the ADDSAFE project.

Some of the important and large European research projects addressing FDD issues are listed in chronological order below.

1. **Fault tolerant flight control activities (2004-2008):** The research was carried out by the GARTEUR(Group for Aeronautical Research and Technology in Europe) Flight Mechanics Action Group FM-AG 16.

2. **The FP7 European Project ADDSAFE (Advanced Fault Diagnosis for Sustainable Flight Guidance and Control,2009-2012):** The research focused on Advanced Fault Detection and Diagnosis towards a more Sustainable Flight Guidance.

3. **The FP7 European Project RECONFIGURE (2013-2015):** The research deals with aircraft GNC technologies that facilitate the automated handling of off-nominal events, funded by the EU FP7.

Note: in the research reported in this thesis no further attention was given to fault detection and diagnosis methods. Instead the following assumption was made:
Assumption 1:

For a certain structural and actuator failures, both the failure types and a rough estimation of the lost quantities can be determined by an FDD unit.

Under this assumption, a pilot is able to switch the flight control system to a suitable control mode to handle the corresponding degraded working condition. For example, the differential thrust control input will only be introduced in the case of a rudder runaway failure or a vertical tail loss case.

2.5. RECONFIGURABLE FLIGHT CONTROL

An overview to the research programs addressing the reconfigurable flight control and the flight control laws that can be used for the purpose of reconfigurable flight control is given in this section.

2.5.1. OVERVIEW OF RECONFIGURABLE FLIGHT CONTROL

A concise overview of reconfigurable flight control approaches and research programs with a focus on accommodating fault scenarios is given here, for more details, the reader is referred to Zhang and Jiang [175], which gives an extensive overview of FTFC approaches with more than 200 references. Note that the overview in this section does not include the fault-tolerant control methods that only deal with sensor failures or switches between redundant hardwares.

Classic reconfigurable flight control (RFC) methods, which can be traced back to the 1980s, require a separate system for FDI. One such methods is that developed by General Electric, where a separate FDI unit is built using an extended Kalman filter. Subsequently, the information from the FDI unit is utilized in the flight control process aimed at realizing an ideal command tracking performance.

Over the past few decades, there have been many research programs in the field of RFC. Some important programs are listed below in chronological order.

1. Self-repairing flight control systems (SRFCS) (1984-1990) [43].
2. Automatic redesign for restructurable control systems (1984-1987) [91].
3. Self-designing flight control for the F-16 VISTA (1993-1996) [99].
4. The X-16 RESTORE program by the US Air Force (1996-2000) [22].
5. Advanced control technology for integrated vehicle (ACTIVE) and intelligent flight control system (IFCS) F-15 program at NASA (1996-2004) [38].
6. Damage tolerant flight control systems for unmanned aircraft by Athena/Honeywell (2007-2008) [49].
7. Fault-tolerant flight control by GARTEUR (2004-2008) [41].

The commonalities and differences between all of the above research programs are summarized in Van Oort [162][pp.6-8] and Lombaerts [86]. For more details of each separate research program or method, the reader is referred to the attached references.

The first example of reconfigurable flight control in a real world problem was that used in a Boeing F/A-18 E/F Super Hornet in 1999 [42]. The designed reconfigurable controller could only accommodate a single stabilator failure mode, where the stabilator was locked in a neutral position.

2.5.2. FLIGHT CONTROL LAWS

A brief overview of the control laws that are suitable for designing a flight controller for an aircraft is given in this subsection. In Subsection 1.2.4, it is stated that the reconfigurable control methods can be classified according to which reconfiguration mechanisms they use, e.g. optimization, switching, matching, model following and compensation [175]. Gain-scheduling control methods and control schemes developed based on linear parameter varying (LPV) models achieve a control reconfiguration by switching between different sets of gains or models. Adaptive control, which includes direct and indirect adaptive control methods, achieves the controller reconfuguration using a compensation mechanism [67].

In practice, gain-scheduling approaches still dominate the control designs of flight control systems of today's aircraft in service. However, in the research reported in this thesis the main focus was placed on relatively modern nonlinear control laws which requires no gain-scheduling procedure. Two nonlinear control laws namely nonlinear dynamic inversion (NDI) and backstepping (BKS) were studied. Furthermore, adaptive control mechanisms were studied. Adaptive control approaches include direct adaptive control and indirect adaptive control. The NDI or BKS control methods become adaptive control methods if certain adaptive mechanisms are introduced.

1. Feedback Linearization

If a nonlinear control system can be transformed into a linear system by a change of coordinates and feedback, we say the system is linearizable. Feedback linearization is also called nonlinear dynamic inversion (NDI), where the control input is chosen in such a way that a nonlinear system is transformed into an equivalent linear system, for which a new virtual control input needs to be further determined. This type of control law can also be further categorized into two groups: input-to-state Linearization and input-to-output linearization. The focus of the research reported in this thesis was input-to-output type of feedback linearization control laws, which are extensively discussed in literature, see [130].

Some authors argue that the stability of a multi-loop controlled system can be guaranteed by using NDI. Others state that NDI cannot guarantee the stability of a multi-loop closed-loop system if the original system cannot be transformed into an equivalent linear system with a companion form by a change of coordinates and feedback.

Despite its numerous advantages relative to classical gain-scheduled flight control, the feedback linearization based flight control approach has two inherent drawbacks. The major drawback of feedback linearization is that it relies on exact cancellation of the nonlinearities in the system, where the model dependency makes the feedback linearization control law highly sensitive to discrepancies between the onboard model and the true aircraft behavior [130, 162]. For small uncertainties, robust control techniques

can be introduced into the controller design to achieve adequate performance. However, if the uncertainties are large, adaptive schemes on top of the basic nonlinear control laws are required, which grants the controller self-tuning capability by adapting the system model in a real-time manner. A second problem of the feedback linearization method is that not all systems can be transformed into a linearized form. Therefore, feedback linearization is not applicable to these systems.

For systems that are not linearizable, quadratic approximate version of feedback linearization controller [70] and dynamic feedback linearization [25] can be used. Alternatively, backstepping technique can also be applied to these type of systems for controller designs.

2. Backstepping

Backstepping is a recursive design method which steps back toward the control inputs starting with the differential equations which are separated from the inputs by the largest number of integrators[71]. Compared to feedback linearization, the backstepping technique is able to address a wider range of controlled systems and allows for more flexibility in the controller design. Firstly, it can deal with a system which can not be transformed to a companion form. Secondly, it enables that stability metrics such as Lyapunov function based stability conditions are incorporated in the controller design.

Lyapunov stability theory is a very important tool for both linear and nonlinear control problems [130, 162]. The recursive backstepping control tool developed for nonlinear control design, along with a Lyapunov function to prove stability of the (multi-loop) closed-loop system, has been well received in the control community. To handle small bounded uncertainties, robust backstepping is developed by introducing nonlinear damping. Other techniques tackling uncertainties include command filtering technique and singular perturbation based backstepping technique [44, 59].

3. Adaptive Control

Research into adaptive flight control can be traced back to the 1950s. The initial objective was to design autopilots for high-performance aircraftoperating at a wide range of speeds and altitudes and thus experiencing large parameter variations [130], and to allow them to fly autonomously. The dynamic behavior of an aircraft depends on its altitude, speed, and configuration. For a fighter aircraft like the F-35, the variations in some modeling parameters may be over 10% during a flight task. For re-entry vehicles, this number may reach 100%. In addition, structural failures such as frame damage can cause large or very large configuration changes. Therefore, adaptive schemes which can make the closed loop aircraft operate in a more optimal or safer state by taking into account all the aforementioned system uncertainties or unpredictable changes is very much needed in the field of aircraft controller design.

Two main configurations for adaptive controls exist: direct adaptive control which directly tunes the controller gains and indirect adaptive control where the tuning gains of the controller are indirectly tuned through adapting the system model. The focus of the research presented in this thesis was the direct control method, in which a controller

is coupled with an independent online estimator of system models.

3-A. Direct Adaptive Control

In the direct adaptive control approach, the system model is not identified and the parameters of the controller are directly updated instead. The adaptation mechanism is designed to adjust the identifier to match some desirable nonlinear controller in such a way that the desired performance of the closed-loop system is achieved.

3-B. Indirect Adaptive Control

In the indirect adaptive control method, an identifier is designed to identify the unknown system dynamics by estimating some model parameters, and the system controller is designed based on *certainty equivalence* control scheme assuming that the estimates of the unknown parameters are their true values. There are two main types of indirect control laws: integral adaptive control and modular adaptive control. In the modular adaptive control approach, a model estimator is designed to update the system model, and it is assumed that the estimates of the model parameters can reach or converge to their true values to allow for the application of the *certainty equivalence* control schemes. The controller part is independent of the model identification part in the gain-tuning mechanism. Unlike modular adaptive control, the integral adaptive control schemes, for example the tuning function based and the Immersion & invariance based adaptive control schemes, tune the system model parameters and the controller parameters in a single cooperated or coupled way. To tolerate system uncertainties, robust techniques such as nonlinear damping and command filtering techniques can be incorporated into both integral adaptive control and modular adaptive control. It should be noted that the weakness of *certainty equivalence* condition will be exposed in the case that big model uncertainties exist or unpredicted dynamics, e.g., the aircraft model configuration changes due to structural or actuator failures.

2.5.3. INCREMENTAL NONLINEAR CONTROL AND SINGULAR PERTURBATION THEORY

1. Incremental Nonlinear Control

A brief introduction on two types of incremental nonlinear control approaches is given in this section: incremental nonlinear dynamic inversion (INDI) and incremental backstepping (IBKS). Due to the similarities between the INDI and IBKS control approaches in the sense of an incremental control concept, only the INDI is introduced in this section. Furthermore, the incremental nonlinear control technique is only discussed here within the scope of aircraft body angular rate control since it is the aerodynamic description of the aircraft rather than the kinematics equations that suffer from disturbances and model uncertainties.

Research into incremental nonlinear dynamic inversion is aimed at removing the model-dependency related drawback of the regular nonlinear dynamic inversion control law. The INDI control scheme requires measurements of angular accelerations and

measured control surface deflections instead of a complete onboard aerodynamic model. In the regular INDI or IB method, the incremental type of expression of the dynamics of the most inner loop is derived by applying a first order Taylor Series expansion to the derivative equations of the body angular rates [44, 129], see Eq. F.7 in Appendix F. Alternatively, incremental nonlinear controller can also be developed using singular perturbation theory [46, 59, 112].

2. Singular Perturbation Theory

Singular perturbation theory is a theoretical advancement that has been made in the control of nonlinear singularly perturbed systems, which have the time scale separation property, see [44, 67]. In [59], a singular perturbation theory based dynamic inversion control approach is developed for a non-affine in control nonlinear system. Falkena et al. [45] have devised an incremental flight controller, which is indicated as a sensor-based backstepping controller, for a small commercial aircraft. More recently, the nonlinear incremental control approach outlined in [45] has been extended in the sense that it has been combined with a backstepping control design technique and validated against disturbances, model uncertainties and sensor noise [44].

2.5.4. CONTROL ALLOCATION METHODS

The flight control laws outlined in the previous section yield desired control action, i.e., moments. The task of control allocation is to distribute a desired control action over the different control effectors available in some optimal way. A control allocation method based on accurate estimation of the aerodynamic model or a single control effectiveness matrix is expected to be able to enhance the performance of FTFC. The quadratic programming control allocation method, which is promising and simple to implement, is briefly discussed in this section. Many other more sophisticated good control allocation methods have also been developed, see [17, 40, 55, 118].

The control allocation problem can be formulated into a quadratic program (QP), which takes magnitude and rate constraints on the control effectors into account. This QP problem can be solved efficiently, so quadratic programming is well suited for on-line applications as an augmentation to the flight control law, for more details, the reader is referred to [162].

The reachable flight envelope of an aircraft is likely to be reduced in the case of a failure. Therefore, the reference signals of an FTFC should also be adapted, i.e., downscaled, accordingly instead of only focusing on control action distribution. This task can be done by introducing pseudo control hedging (PCH) [63, 64], which can compensate for the input characteristics such as actuator position limits, actuator rate limits and linear dynamics of the actuator. In [57], PCH is incorporated into an adaptive flight control setup based on nonlinear dynamic inversion.

However, PCH, which can approximately make a fast adjustment on the reference commands, is only based on the capability or dynamics of the available control effectors. In the past few decades, many different methods for safe-flight-envelope prediction and protection have been developed and flight envelope prediction has become an independent research field. Flight envelope prediction methods, which takes into account

the control surface availability and output of the closed-loop system simultaneously, are introduced in the next section.

2.6. FLIGHT ENVELOPE PROTECTION

All flight vehicles have a high safety level requirement compared to ground vehicles [58, 85, 127, 164, 171]. A large percentage of the loss of control in flight (LOC-I), which is the most frequent major cause of fatal accidents [12, 85, 151], can possibly be prevented from happening or be recovered from in some accident cases using modern guidance and control techniques [85, 127]. An overall FTFC system should be able to detect and tolerate or adapt to the changes in the aircraft behavior following an accident. In the research reported in this thesis the focus was placed on model-based FTFC systems defined in Section 1.2, e.g., a physical aerodynamic model was used to design an adaptive reconfigurable controller. Within the framework of FTFC, two main branches exist aimed at enhancing flight safety, namely non-conventional reconfigurable control laws, and flight envelope static or dynamic protection. Flight envelope protection is aimed at preventing an aircraft, which is either in a highly maneuvering flight condition or a post-failure case, from entering an upset flight condition by providing the pilot, or an internal controller, with current achievable operating commands. In this thesis, the flight envelope protection unit is treated as a block which is viewed as being parallel to the reconfigurable control laws unit, see Figure 2.1.

Flight envelope protection can be categorized into two branches: static and dynamic, i.e., online. The approaches for static envelope estimation include wind tunnel testing, real flight test experiments and high-fidelity model-based computation of attainable equilibrium sets or achievable trim points [19, 52, 66, 149]. After getting a safe envelope set, a high dimensional envelope model can be constructed or expanded using regular function approximation approaches such as polynomials, multivariate splines and neural networks.

Online flight envelope protection (OFEP) has gained increasing attention from the aerospace research community. For the purpose of OFEP, the approaches based on the analysis of the forward reachability and backward reachability are the preferred methods due to the intuitive physical interpretation of the interception safe set [54, 85, 127, 164]. In this type of methods, the OFEP problem is formulated into a reachability problem, which is called mathematically a *Bolza problem* [11, 54], i.e., a continuous-time optimal control problem in a fixed, finite time horizon, where the associated HJB equations with a time-dependent integral cost function are solved in real-time [54, 60, 127, 164].

The Hamilton-Jacobi PDEs are commonly solved using level set methods [162] along with other PDE solvers such as finite element methods, but these grid-based evolution methods suffer from the *curse-of-dimensionality*. For OFEP, one major challenge is the high computational load introduced by the evolution process of the Hamilton-Jacobi PDEs. The challenge becomes even bigger if the system has a high number of dimensions, e.g., higher than 5. One solution to reduce the computational load is to take advantage of the time scale separation property of the aircraft system and simplify a high-order aircraft system description into multiple low-order subsystems. Alternatively, the development of more efficient mathematical tools can possibly be pursued in the future to solve Hamilton-Jacobi PDEs more efficiently. For example, the

max-plus method [95] provides a curse-of-dimensionality-free variant. However, this state-of-the-art mathematical tool is not yet sufficiently mature and is far from ready for a real life application.

Another alternative to the aforementioned methods for solving the *Bolza problem* exists. To solve the HJB equations efficiently without suffering from the *curse-of-dimensionality*, an alternative commonly used method is to approximate the cost function using generic function approximation methods such as reinforcement learning methods [14, 122]. Using reinforcement learning approaches such as radial basis functions neural networks (RBF-NN) or MVSB, the approximated cost function of the *Bolza problem* can be updated using a collocation method using training datasets, which have a sufficient coverage of the entire flight envelope [4, 54, 60]. Huang et al. [60] have developed a collocation method to solve the HJB equations using RBF-NN. This method has been extended by Alwardi et al. [4] with an adaptive algorithm, which refines the distribution of the RBF centers based on the feedback information of the approximation errors. In [54], Govindarajan et al. present a collocation method for solving time-dependent HJB equations using a sparse functional approximation algorithm called MVSB, the sparsity of which contributes to the decrease of the computational load.

Finally, it should not be ignored that a prerequisite of all the OFEP methods and the model-based static flight envelope protection methods is to achieve an accurate global valid aerodynamic model which can reflect the ongoing changes in an aircraft's flight dynamics by capturing the changes in the aircraft structural or actuator configurations taking into account possible damage.

I

GLOBAL AERODYNAMIC MODEL IDENTIFICATION

This part focuses on developing a new type of recursive methods for identifying a global aerodynamic model, which is required for the work of model-based adaptive control and model-based flight envelope protection. In a model-based flight control system, the accuracy of an aerodynamic model might directly affects the command tracking performance of the closed-loop aircraft system. The use of a global aerodynamic model enables the estimated model to be stored for later reuse. If the same flight condition is revisited, the model needs not to be estimated for a second time. Many static flight envelope protection approaches and most of the online flight envelope protection methods rely on an accurate aerodynamic model which is valid over the entire flight envelope. In online flight envelope protection, the dynamic estimation of a safe-flight-envelop is usually based on the evolution of a reliable and accurate global valid aerodynamic model.

The preferred aircraft model identification method in this thesis is a two-step method. This method consists of two consecutive procedures: a state estimation process using an extended Kalman filter and a parameter identification process using a function approximation algorithm. Given an accurate estimation of the states, the overall aircraft model identification problem becomes an *equation-error* identification problem, which can be solved using regular function approximation approaches.

In part one of this thesis, three new recursive type identification methods which can provide precise global aerodynamic models are presented. They are shown in Chapter 3, Chapter 4 and Chapter 5, respectively.

The identification methods using multivariate simplex B-splines are one of the main focus in part one. This type of methods has many advantages over other methods. Firstly, unlike ordinary tensor product splines, simplex B-spline based method can deal with scattered data sets without needing pre-treatment. Secondly, simplex B-spline based method has proven to have a high approximation power. Its approximation power can be increased by increasing the density of the simplices and the polynomial order. Thirdly, this method can guarantee a smooth transition between different local per-simplex models. Fourthly, this method allows interpolation between simplices or extrapolation outside of the well-studied subdomains with a boundary predetermined by the *a priori* knowledge. In addition, Directional derivative can be easily calculated. Chapter 3 and Chapter 4 developed two efficient parameter estimation algorithms for updating the spline model in a recursive manner.

Recursive kernel methods have also proven to be a powerful model identification tool, and have called the attention from many different research communities and industry fileds. Using a nonlinear mapping through kernel functions, a kernel method is able to transform a nonlinear function in an original input space into a linear function in the kernel space. An open issue for recursive kernel methods is how to choose an optimum or optimal set of kernels, which should be representing and the number of which should be as as possible, in a computationally efficient way. In Chapter 5, an improved recursive reduced least squares support vector regression method is used to provide kernel centers for a classical recursive kernel method. In addition, to better capture the local data trends, the benefits of expanding the local kernels are investigated.

3

SUBSTITUTION BASED RECURSIVE MULTIVARIATE SIMPLEX B-SPLINES

In the previous chapter, a literature review of each component in an overall model-based fault-tolerant flight control system was given. A conclusion was drawn that an accurate global aerodynamic model played an important role in designing a model-based reconfigurable controller and a flight envelope protection unit. Using a two-step method, aircraft model identification turns into *equation error* model identification problem given accurate aircraft states. An *equation error* model identification problem can be solved using many different function approximation approaches such as neural networks, polynomial based methods and multivariate simplex B-spline based method. The model structure of neural networks is non-transparent, and neural networks commonly suffers from a parameter convergence problem. The polynomial based method is commonly subject to an overfitting problem. An alternative to the aforementioned methods is the multivariate simplex B-splines (MVSB), which has received increasing attention from the system identification field. The MVSB method has the potential to lead to a higher accuracy than a regular model identification method based on polynomial basis.

Model identification includes two parts of work: the abovementioned model structure selection and parameter estimation. This chapter focuses on parameter estimation. With regard to recursive model identification methods using simplex B-splines, the equality constrained recursive least squares (ECRLS) MVSB method is hard to apply in real time when the number of B-coefficients is high, its computational load still needs to be reduced to facilitate on-board applications. In this chapter, a substitution strategy based on singular value decomposition is proposed to speed up a recursive MVSB method.

A new substitution based (SB) recursive identification method, using multivariate simplex

B-splines (MVSBs), has been developed for the purpose of reducing the computational time in updating the spline B-coefficients. Once the structure selected, the recursive identification problem using the MVSBs turns to be a constrained recursive identification problem. In the proposed approach, the constrained identification problem is converted into an unconstrained problem through a transformation using the orthonormal bases of the kernel space associated with the constraint equations. The main advantage of this algorithm is that the required computational time is greatly reduced due to the fact that the scale of the identification problem, as well as the scale of the global covariance matrix, is reduced by the transformation. For validation purpose, the SB-RMVSBs algorithm has been applied to approximate a wind tunnel data set of the F-16 fighter aircraft. Compared with the batch MVSBs method and the equality constrained recursive least squares (ECRLS) MVSBs method, the computational load of the proposed SB-RMVSBs method is much lower than that of the batch type method while it is comparable to that of the ECRLS-MVSBs method. Moreover, the higher the continuity order is, the less computational time the SB-RMVSBs method requires compared with the ECRLS-MVSBs method.

3.1. INTRODUCTION

The control performance of a model-based automatic control system, like for example the adaptive nonlinear dynamic inversion (ANDI) flight control system [87, 163] and the module based adaptive backstepping flight control system [163], heavily relies on the accuracy of the object model that is identified in real-time. Recently, de Visser et al. [34] proposed a novel batch type identification method using multivariate simplex B-splines. Comparing with the ordinary polynomial basis (OPB) based method, this simplex spline basis (SSB) based method can provide a relatively more stable basis and enjoys a higher approximation power owing to the fact that multiple local modules are identified instead of identifying a single overall model[33]. Another main merit of the multivariate simplex B-splines (MVSBs) is that they are capable of using the scattered dataset as training data. This is a property that the multivariate sensor product splines method does not have [34].

Later, de Visser and Chu et al. [37] developed an equality constrained recursive least squares (ECRLS) based MVSBs identification method after combining the linear regression formulation of the spline bases from [34] with the recursive least squares identification method from [180]. The recursive identification method presented in [180] can convert a constrained identification problem into a free-of-constraint identification problem . In this recursive identification method, the constrained recursive identification process is circumvented by merely injecting the equality constraint information into the general least square solution calculated using an initial training data collection.

However, in order to enable the real-time aerodynamic model identification, it is still necessary to reduce the computational load of the recursive MVSBs method. This chapter is aimed at providing a more effective recursive identification method than the ECRLS-MVSBs method developed in [37]. The new method should enjoy a much lower computational load than the batch MVSBs, and have a lower computational load than the ECRLS-MVSBs method. In this chapter, a new substitution based multivariate simplex B-splines (SB-MVSBs) method is developed. The kernel-space bases based

transformation can greatly cut down the computational time required by the SB-MVSBs method.

This chapter is structured as follows. The preliminaries on the multivariate simplex B-splines are introduced in Section 3.2. The SB-MVSBs method is developed in Section 3.3. In Section 3.4, the proposed SB-RMVSBs method is applied to a wind tunnel data set of the F-16 fighter aircraft, and the selection of the spline function structure is investigated. Subsequently, the proposed method is compared with both the batch method and the ECRLS-MVSBs method in Section 3.5. Finally, this chapter is concluded by Section 3.6.

3.2. PRELIMINARIES ON MULTIVARIATE SIMPLEX B-SPLINES

The basic principles for simplex splines are briefly introduced in this section. Without this introduction, the formulation of the SB-MVSBs method will be incomplete.

3.2.1. SIMPLEX AND BARYCENTRIC COORDINATES

Let t be an n-simplex formed by the convex hull of its $n+1$ non-degenerate vertices $(v_0, v_1, ..., v_n) \subset \mathbb{R}^n$. The normalized barycentric coordinates of some evaluation point $\mathbf{x} \in \mathbb{R}^n$ with respect to simplex t are defined as

$$b(\mathbf{x}) \quad := \quad (b_0, b_1, ..., b_n) \in \mathbb{R}^{n+1}, \ \mathbf{x} \in \mathbb{R}^n \tag{3.1}$$

which follows from the following implicit relation:

$$\mathbf{x} = \sum_{i=0}^{n} b_i v_i, \ \sum_{i=0}^{n} b_i = 1 \tag{3.2}$$

3.2.2. TRIANGULATIONS OF SIMPLICES

The approximation power of the multivariate simplex spline is partly determined by the structure of the triangulation. A triangulation $\mathcal{T}$ is a special partitioning of a domain into a set of J non-overlapping simplices:

$$\mathcal{T} := \bigcup_{i=1}^{J} t_i, \ t_i \cap t_j \in \{\varnothing, \tilde{t}\}, \forall \ t_i, t_j \in \mathcal{T} \tag{3.3}$$

with the edge simplex $\tilde{t}$ a k-simplex with $0 \leq k \leq n-1$. High quality triangulations can be obtained using constrained Delaunay triangulation (CDT) methods, such as the 2-dimensional CDT method presented by Shewchuk [128].

3.2.3. BASIS FUNCTIONS OF THE SIMPLEX B-SPLINES

According to [73] and [34], the Bernstein basis polynomial $B_\kappa^d(b(\mathbf{x}))$ of degree d in terms of the barycentric coordinates $b(\mathbf{x}) = (b_0, b_1, ..., b_n)$ from Eq. 3.2 is defined as:

$$B_\kappa^d(b(\mathbf{x})) := \begin{cases} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} & , \mathbf{x} \in t \\ 0 & , \mathbf{x} \notin t \end{cases} \tag{3.4}$$

where $\kappa = (\kappa_0, \kappa_1, ..., \kappa_n) \in N^{n+1}$ is a *multi-index* with the following properties: $\kappa! = \kappa_0!\kappa_1!...\kappa_n!$ and $|\kappa| = \kappa_0 + \kappa_1 + ... + \kappa_n$. In Eq. 3.4 we use the notation $b^\kappa = b_0^{\kappa_0} b_1^{\kappa_1} ... b_n^{\kappa_n}$. Given that $|\kappa| = d$, the total number of valid permutations of the *multi-index* κ is:

$$\hat{d} = \frac{(d+n)!}{n!d!} \tag{3.5}$$

In [30], it was proved that any polynomial $p(b)$ of degree d on a simplex t can therefore be written as a linear combination of $\hat{d}$ basis polynomials in what is known as the B-form as follows:

$$p^t(b(\mathbf{x})) := \begin{cases} \sum_{|\kappa|=d} c_\kappa^t B_\kappa^d(b(\mathbf{x})) & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases} \tag{3.6}$$

with c_κ^t the B-coefficients which uniquely determines $p^t(b(\mathbf{x}))$, where the superscript 't' indicates that p is defined on the simplex 't'. The total number of basis function terms is equal to $\hat{d}$, which is the total number of valid permutations of κ.

3.2.4. VECTOR FORMULATIONS OF THE B-FORM

As introduced in [37], the vector formulation, according to Eq. 3.6, for a B-form polynomial $p(b(\mathbf{x}))$ in barycentric $\mathbb{R}^{n+1}$ has the following expression:

$$p^t(\mathbf{x}) := \begin{cases} \mathbf{B}_t^d(b(\mathbf{x})) \cdot \mathbf{c}^t & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases}, \tag{3.7}$$

with $b(\mathbf{x})$ the barycentric coordinates of the Cartesian $\mathbf{x}$. The row vector $\mathbf{B}_t^d(b(\mathbf{x}))$ in Eq. 3.7 is constructed from individual basis polynomials which are sorted lexicographically[37].

The simplex B-spline function $s_d^m(b(\mathbf{x}))$ of degree d and continuity order m, defined on a triangulation consisting of J simplices, is defined as follows:

$$s_d^m(\mathbf{x}) := \mathbf{B}^d(b(\mathbf{x})) \cdot \mathbf{c} \ \in \mathbb{R}, \tag{3.8}$$

with $\mathbf{B}^d(b(\mathbf{x}))$ the global vector of basis polynomials which has the following full expression:

$$\mathbf{B}^d(b(\mathbf{x})) := [\ \mathbf{B}_{t_1}^d(b(\mathbf{x})) \quad \mathbf{B}_{t_2}^d(b(\mathbf{x})) \quad \cdots \quad \mathbf{B}_{t_J}^d(b(\mathbf{x}))\ ] \in \mathbb{R}^{1 \times J \cdot \hat{d}} \tag{3.9}$$

Note that according to Eq. 3.7 we have $\mathbf{B}_{t_j}^d(b(\mathbf{x})) = 0$ for all evaluation locations $\mathbf{x}$ that are located outside of the triangle t_j. This results in that $\mathbf{B}^d$ is a sparse row vector.

The global vector of B-coefficients $\mathbf{c}$ in Eq. 3.8 has the following formulation:

$$\mathbf{c} := \begin{bmatrix} \mathbf{c}^{t_1\top} & \mathbf{c}^{t_2\top} & \cdots & \mathbf{c}^{t_J\top} \end{bmatrix}^\top \in \mathbb{R}^{J \cdot \hat{d} \times 1} \tag{3.10}$$

with each $\mathbf{c}^{t_j}$ a per-simplex vector of lexicographically sorted B-coefficients.

For a single observation on y we have:

$$f = \mathbf{B}^d(b(\mathbf{x}))\mathbf{c} + \varepsilon \tag{3.11}$$

with ε the residue. Then, for all the N observations, we have the following well-known formulation:

$$\mathbf{f} = \mathbf{X}(b(\mathbf{x}))\mathbf{c} + \xi \in \mathbb{R}^{N \times 1} \tag{3.12}$$

with $\mathbf{X}(b(\mathbf{x})) \in \mathbb{R}^{N\times J\cdot\hat{d}}$ a collection matrix of the row vector $\mathbf{B}^d$ from Eq. 3.9, and $\xi = [\varepsilon_1, \varepsilon_2, ...\varepsilon_N]^T$ the residue vector. For writing convenience, $\mathbf{X}(b(\mathbf{x}))$ will be written as $\mathbf{X}$ in the remainder of this chapter.

3.2.5. GLOBAL CONTINUITY CONSTRAINTS

The formulation for the continuity conditions from [10] and [73] is used:

$$\mathbf{c}^{t_i}_{(\kappa_0,...,\kappa_{n-1},m)} = \sum_{|\gamma|=m} \mathbf{c}^{t_j}_{(\kappa_0,...,\kappa_{n-1},0)+\gamma} B^m_\gamma(\mathbf{v}), \quad 0 \le m \le r \tag{3.13}$$

with $\mathbf{v}$ the Bernstein coordinates of the vertex which only belongs to the i^{th} simplex, $\gamma = (\gamma_0, \gamma_1, ..., \gamma_n)$ a multi-index independent of κ, $|(\kappa_0, ..., \kappa_{n-1}, m) + \gamma| = d$. t_i, t_j denote the i-th and j-th simplices separately.

Eventually, the following equality constraints should be maintained during the calculation of the global B-coefficient vector $\mathbf{c}$:

$$\mathbf{H}\cdot\mathbf{c} = 0 \tag{3.14}$$

with $\mathbf{H} \in \mathbb{R}^{(E\cdot R)\times(J\cdot\hat{d})}$ the smoothness matrix [34], R is the number of continuity conditions per edge. E is the number of edges in the specified triangulation. If all the simplices' surfaces connect smoothly on the edges within the whole triangulation, we call the simplex splines globally continuous. Global continuity is determined by Eq. 3.13 and Eq. 3.14.

3.2.6. SPLINE FUNCTION SPACE AND A POLYNOMIAL FUNCTION SPACE

In this chapter, we use a new type of definition of polynomial function space:

$$P_d(n) := \{p_k(\mathbf{x}) : p_k|_{\mathbf{x}} \in \mathbb{P}_k, \ \forall \mathbf{x} \in \mathbb{R}^n \ and \ \forall k \le d\} \tag{3.15}$$

with $\mathbf{x}$ the input vector, $\mathbb{P}_k$ the space of polynomials of degree k.

We use the following definition of the spline space, which is a modified form of the definition given by Lai et al. in [73]:

$$S^m_d(n) := \{s^m_d(\mathbf{x}) \in C^m : s^m_d|_{\mathbf{x}} \in \mathbb{P}_d, \ \forall \mathbf{x} \in \mathbb{R}^n\} \tag{3.16}$$

with $\mathbb{P}_d$ the space of polynomials of degree d, and n the dimension of function inputs.

Note that, the former represents the ordinary polynomial function bases with the order up to d. For example, if we select $\mathbf{x} = [x, \ y]^T$, then $P_2(2) := c_1 + c_2 x + c_4 y + c_3 x^2 + c_6 xy + c_5 y^2$ with x and y two elements of $\mathbf{x}$.

3.3. TRANSFORMATION BASED RECURSIVE IDENTIFICATION METHOD

The kernel space information of the equality constraint matrix $\mathbf{H}$, formulated in Eq. 3.14, has been utilized to transform the constrained recursive identification problem into a free-of-constraint recursive identification problem.

3.3.1. TRANSFORMATION OF CONSTRAINTS

Once the triangulation and the spline function structure are chosen, the equality constraints have the property that they are time invariant and known a priori. In this case, a straightforward substitution method can be applied to remove the constraints in each recursion step.

Following from Eq. 3.8, the original constrained recursive identification problem has the following expression:

$$f = \mathbf{B} \cdot \mathbf{c} + \varepsilon \tag{3.17}$$

$$\text{s.t. } \mathbf{H} \cdot \mathbf{c} = 0 \tag{3.18}$$

Assume that the singular value decomposition (SVD) result of $\mathbf{H}$ is as follows:

$$\mathbf{H}_{n\times m} = \mathbf{V}_{n\times n} \begin{bmatrix} \Sigma_{r\times r} & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(m-r)} \end{bmatrix} \mathbf{U}^T_{m\times m} \tag{3.19}$$

where $\Sigma = \text{diag}\begin{pmatrix}\sigma_1 & , & \cdots & \sigma_r\end{pmatrix}$ is the diagonal vector of all singular values , $\sigma_1 \geq \cdots \geq \sigma_r > 0$ and r is the rank of $\mathbf{H}$. $\mathbf{V} = \begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix}$ is an n_{th} order orthogonal matrix, $\mathbf{V}_1$ is an n by r matrix. $\mathbf{U} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}$ is a m_{th}-order orthogonal matrix, $\mathbf{U}_1$ is an m by r matrix. Because $\mathbf{c} \in null(\mathbf{H})$, one feasible general solution for the homogeneous equation Eq. 3.18 is:

$$\mathbf{c} = \mathbf{U}_2\mathbf{y} \tag{3.20}$$

where the column vectors of $\mathbf{U}_2$ form an orthonormal basis of $null(\mathbf{H})$ [96, 172]. $\mathbf{y}$ is a column vector which needs to be calculated (identified) later, and its length is $m-r$. The feasibility of the above mentioned conversion will be proved later in theorem 1.

By introducing this general solution into Eq. 3.17, we get the following formation:

$$f = \mathbf{B}\mathbf{U}_2\mathbf{y} + \varepsilon \tag{3.21}$$

with $\mathbf{U}_2$ a basis for $null(\mathbf{H})$. Since Eq. 3.21 only represents an unconstrained identification problem, a regular recursive least squares identification method becomes capable to solve it. In order to obtain the final unknown parameters (B-coefficients), we only need to substitute the identified vector $\mathbf{y}$ into Eq. 3.20. The computational flow chart is concluded as follows.

Algorithm 1:

step.1 determine the triangulation $\mathcal{T}$, calculate the smoothness matrix $\mathbf{H}$, and carry out the SVD according to Eq. 3.19 to get $\mathbf{U}_2$.

step.2 calculate the spline basis vector according to Eq. 3.9.

step.3 identify the unknown vector $\mathbf{y}$ contained by Eq. 3.21 using a regular recursive least squares method.

step.4 reconstruct the B-coefficient vector $\mathbf{c}$ from the vector $\mathbf{y}$ using Eq. 3.20. Return to step.2 if a new data is available.

Theorem 1: Optimal approximation

Given $\mathbf{y}$ the unique and optimal least square estimation vector of problem Eq. 3.21, $\mathbf{c} = \mathbf{U}_2\mathbf{y}$ is the optimal least squares solution of the constrained problem Eq. 3.17.

Proof:

Given $\mathbf{U}_2$ derived from Eq. 3.19, columns of matrix $\mathbf{U}_2$ constitute orthonormal bases for the kernel space of $\mathbf{H}$. Therefore, we have $\mathbf{H}\mathbf{U}_2 = 0$. Hence, we can get $\mathbf{H}\mathbf{U}_2 \cdot \mathbf{y} = 0$. Because $\mathbf{c} = \mathbf{U}_2\mathbf{y}$ as shown in Eq. 3.20, we can get $\mathbf{H} \cdot \mathbf{c} = 0$. The equality constraints $\mathbf{H} \cdot \mathbf{c} = 0$ are satisfied during parameter estimation.

Because Eq. 3.17 and Eq. 3.21 hold, we have

$$\mathbf{f} - \mathbf{X} \cdot \mathbf{c} = \xi = \mathbf{f} - \mathbf{X}\mathbf{U}_2\mathbf{y} \tag{3.22}$$

We define the cost function of the least square problem as $C(\mathbf{c}) = \min_{\mathbf{c}} \xi^T \xi$, where $\mathbf{c}$ is the vector to estimate. As $\mathbf{y}$ is the optimal and unique least square solution of problem 3.21, we assume that it leads to a minimum residual vector ξ_d, so the minimum cost function value can be written as $C(\mathbf{y}) = \xi_d^T \xi_d$. Because the two problems described by Eq. 3.17 Eq. 3.18 and Eq. 3.21 are identical systems in view of the output approximation, we can get the following result: $C(\mathbf{c}) = C(\mathbf{y}) = \xi_d^T \xi_d$ from Eq. 3.22. □

3.3.2. REMARKS

Note that, according to Eq. 3.21, the proposed recursive identification method has cut down the scale of the original identification problem by multiplying the regression data matrix by $\mathbf{U}_2$ from the right hand side.

There exist some similarities between the SB-MVSBs method and the orthogonal least squares based identification method presented in [137]. In theory, the singular value decomposition allows to reduce the structure of the aerodynamic model. By keeping all (non-zero) singular values, the SB-MVSBs method has removed the dependent columns in the data matrix. However, it is not reasonable to cut out the smallest singular values and further reduce the scale of the model because the constraints are originally added to the unknown parameters rather than to the regression data matrix.

3.4. VALIDATION USING WIND TUNNEL DATA OF THE F-16 FIGHTER AIRCRAFT

3.4.1. F-16 AERODYNAMIC MODEL STRUCTURE

According to the F-16 aerodynamic wind tunnel data presented in [114], the following structure is a good option for X-direction aerodynamic force (moment) coefficient:

$$\begin{aligned} F_x\left(\alpha, \beta, \delta_e, \delta_{lef}, \frac{q\bar{c}}{V}\right) &= f_1(\alpha, \beta, \delta_e) + f_2(\alpha, \beta) \cdot \delta_{lef} \\ &+ f_3(\alpha) \cdot \frac{q\bar{c}}{V} + f_4(\alpha) \cdot \frac{q\bar{c}}{V} \cdot \delta_{lef} \end{aligned} \tag{3.23}$$

Note that the engine thrust is assumed to be constant and its related term is removed from Eq. 3.23. According to Eq. 3.23, once the q, V and δ_{lef} are fixed, we can derive the following linear regression formulation for a three dimensional MVSBs function.

$$S(x) = \mathbf{B} \cdot \mathbf{c} \tag{3.24}$$

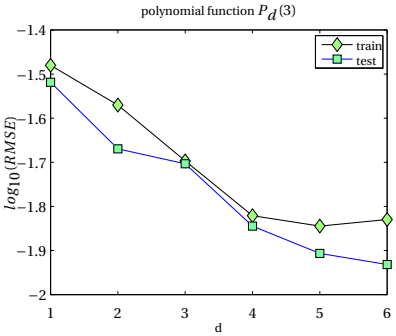


Figure 3.1: Different selection of d for $P_d(3)$, C_m.

where **B** is the B-form spline vector calculated using Eq. 3.9.

According to de Visser [35], the global continuity matrix **H** for the three dimensional MVSBs function should be calculated using Eq. 3.13.

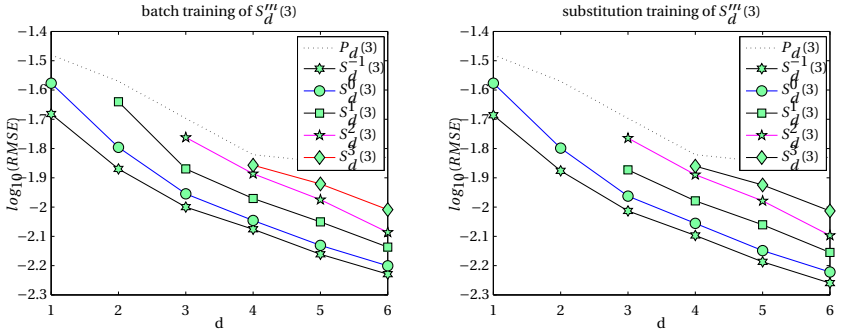
In the simulation, an aerodynamic model of the F-16 aircraft was identified using simulated flight test data generated with a nonlinear F-16 simulator based on a NASA wind tunnel dataset [114]. The training inputs of the simulated flight test dataset were obtained by generating 20,000 uniformly distributed inputs within their own valid regions. The inputs of the test dataset containing 4331 points are produced by the grids determined by α and β. The system output were calculated through the high resolution interpolation from the wind tunnel data provided by [114] with $\delta_{lef} = 1^\circ$, $V = 600\,ft/s$, $q = 0.1\,rad/s$, $\bar{c} = 11.32\,m$. Moreover, the model outputs of the aerodynamic model is contaminated artificially by adding a white noise with a magnitude of 1% (relative to its maximum and minimum value).

3.4.2. CROSS VALIDATION RESULTS IN DETERMINING THE STRUCTURE

In the numerical simulation, we have chosen the MVSBs function to have only one three dimensional sub-function. The notation $S_d^m(n)$ from Section 3.2 has been used, and the overall spline function becomes the following expression:
$S(x) = S_d^m(n)$, where $n = 3$, while d, m are kept undetermined. The partitioning vector of α is [−20 10 40]. The partitioning vector of β is [−25 25]. The partitioning vector of δ_e is [−20 20]. In order to enhance the approximation ability of this algorithm, all the inputs are normalized into the closed range of [0 1]. In order to select a suitable structure for the spline model of C_m, i.e., the nondimensional pitch moment coefficient, the effects of the structural parameters, i.e., d and m, will be investigated. To demonstrate the approximation power of the SB-MVSBs method, we compared it with the batch MVSBs method.

Figure 3.1 shows the root mean squared errors (root mean squared errors (RMSE)) of the fitting outputs (C_m) using the ordinary polynomial basis (OPB) based recursive least squares identification method.

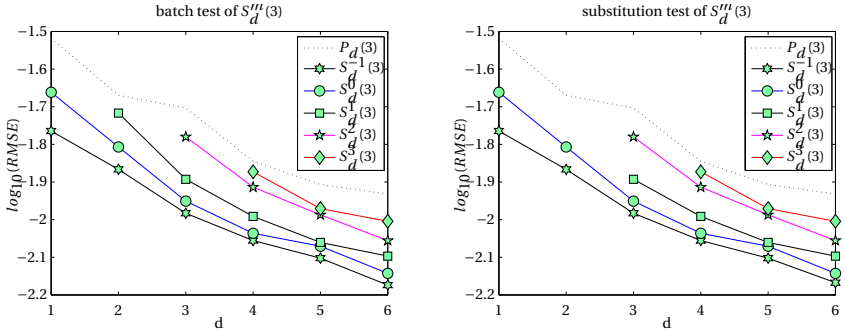
Figure 3.2(a) and Figure 3.2(b) show the RMSE of the training data set using the



(a) batch MVSBs training. (b) SB-MVSBs training.

Figure 3.2: Different combination of m and d for $S_d^m(3)$, $\mathcal{T}_{12}$, C_m.

batch MVSBs method and the proposed SB-MVSBs method respectively. Comparing these two figures, it has been found that the SB-MVSBs method enjoys the same level of approximation power as that of the batch MVSBs.



(a) batch MVSBs test. (b) SB-MVSBs test.

Figure 3.3: Different combination of m and d for $S_d^m(3)$, $\mathcal{T}_{12}$, C_m.

Figure 3.3(a) and Figure 3.3(b) show the RMSE of the testing data set based on the B-coefficients identified using the batch MVSBs method and the SB-MVSBs method respectively. As can be seen from these two figures, the approximation power of the batch MVSBs method and the SB-MVSBs method are very close. Moreover, compared with the results shown in Figure 3.1, Figure 3.3 indicate that both the batch MVSBs method and the SB-MVSBs method enjoy a higher approximation power than the OPB based recursive identification method.

Table 3.1: Computational Complexity (CC) in time

Methods	batch MVSBs	ECRLS-MVSBs	SB-MVSBs
CC	$\mathcal{O}\left(m^3\right)$	$\mathcal{O}\left(3m^2\right)$	$\mathcal{O}\left\{(m-r)\cdot(4m-3r)\right\}$

Table 3.2: Computational time for 20k data of C_m, $\mathcal{T}_{12}$, B-coefficient number 1008, $S_6^m(3)$

condition	$S_6^{-1}(3)$	$S_6^0(3)$	$S_6^1(3)$	$S_6^2(3)$	$S_6^3(3)$	$S_6^4(3)$
ECRLS	104.5092	105.5291	105.0324	106.2780	106.2854	106.6970
SB-MVSBs(operated)	101.7709	33.2808	13.4270	5.3797	4.2410	3.7263
SB-MVSBs(normal)	139.4835	67.7009	24.0644	7.9068	6.0565	5.6464

3.5. COMPARISON WITH THE ECRLS-MVSBS AND THE BATCH MVSBS

3.5.1. COMPUTATIONAL COMPLEXITY

The computational complexity of the substitution based MVSBs (SB-MVSBs) method is split into two parts. Firstly, according to Eq. 3.21, the multiplication between the $\mathbf{B}$ vector and the $\mathbf{U}_2$ matrix needs $m\cdot(m-r)$ with r the rank of the continuity matrix, and m the length of the B-coefficient vector $\mathbf{c}$. Similar to the ECRLS method, the computational complexity for the pure regression process using the recursive least squares is $\mathcal{O}\left(3(m-r)^2\right)$. By summing them up, we can get the total computational complexity of the SB-MVSBs method: $C(m,r)=(m-r)\cdot(4m-3r)=3r^2-7mr+4m^2$. The computational complexity in time of the batch MVSBs method, the ECRLS-MVSBs method and the SB-MVSBs method are tabulated in Table 3.1.

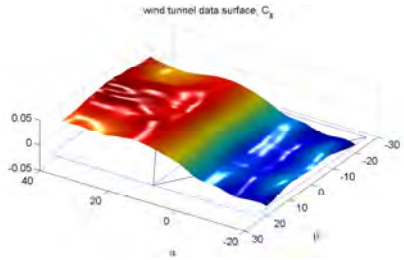
Given m, function $C(m,r)$ monotonously increases as $r<m$. Therefore the minimum computational complexity of the SB-MVSBs method is $4m^2$ when $r=0$, while its highest limit is 0. In addition, $C(m,r)=3n^2$ holds when $r=\frac{(7-\sqrt{37})}{6}m$.

3.5.2. COMPUTATIONAL TIME COMPARISON WITH THE ECRLS-MVSBS

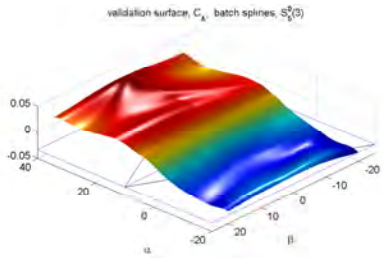
In order to reveal the influence of the continuity order m on the computational complexity in time, a numerical experiment is performed with different selection for the continuity order m. In the remainder of this chapter, we will always choose the MVSBs function to have only one three dimensional sub-function in all of the numerical experiments. The simulation results are listed in Table 3.2. In Table 3.2, '$operated$' means that the $\mathbf{BU}_2$ multiplication shown in Eq. 3.21 is executed in advance in a batch manner. According to Table 3.2, the SB-MVSBs method require less computational time than the ECRLS-MVSBs method, and this advantage will become more apparent with the increase of the continuity order m.

3.5.3. EVALUATION RESULTS ON THE APPROXIMATION POWER

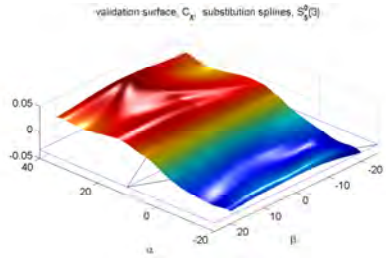
The OPB based recursive identification method, the batch MVSBs method and the SB-MVSBs recursive identification method are utilized to fit the same training data set of C_x respectively. The models identified using these three different methods respectively



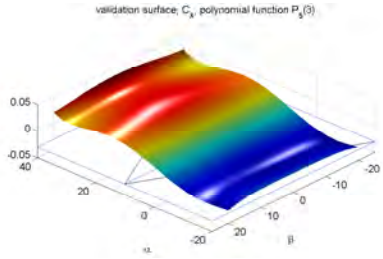
(a) Original wind tunnel data surface.



(b) batch spline function surface.



(c) substitution spline function surface.



(d) Polynomial fitting surface.

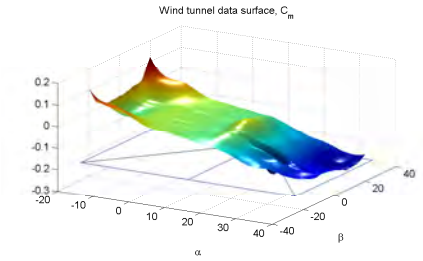
Figure 3.4: Validation surface of C_x ($\delta_e = 2^o$), $\mathcal{T}_{12}$.

are validated using the testing data that are located on the mesh grids. The validation surfaces of C_x are shown in Figure 3.4. Apparently, the SB-MVSBs method enjoys an equal fitting accuracy to that of the batch MVSBs method while having a higher approximation power than the OPB based recursive identification method.

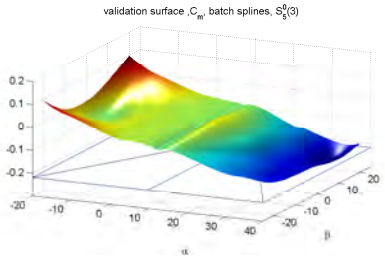
The OPB based recursive identification method, the batch MVSBs and the SB-MVSBs recursive identification methods are utilized to fit the same training data set of C_m. The models identified using three different methods are validated using the same testing data set as that mentioned previously. The validation surfaces of C_m are plotted in Figure 4.9. We can get a similar conclusion as that drawn from last experiment that the SB-MVSBs method has the same fitting power as the batch MVSBs method while having a higher approximation power than the OPB based recursive identification method.

3.6. CONCLUSIONS

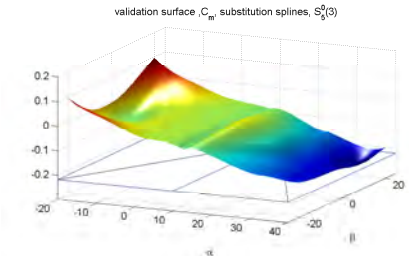
A new substitution based recursive MVSBs method is proposed for the online aerodynamic model identification. In view of the equality constraints contained by the MVSBs, a SVD based transformation is empoyed to convert an originally constrained recursive identification problem into a free-of-constraint identification problem. The proposed recursive model identification method namely SB-MVSBs method was applied to approximate a series of two wind tunnel data sets of F-16 aircraft, and were compared with the batch MVSBs method and the ECRLS-MVSBs method. The numerical simulation results show that the proposed SB-MVSBs method requires less computational



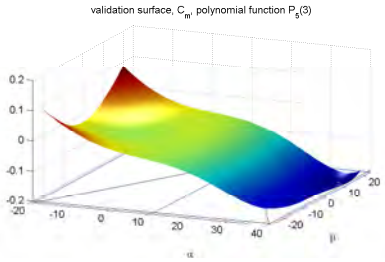
(a) Original wind tunnel data surface.



(b) batch spline function surface.



(c) substitution spline function surface.



(d) Polynomial fitting surface.

Figure 3.5: Validation surface of C_m ($\delta_e = 2^o$), $\mathcal{T}_{12}$.

time than the batch MVSBs method and the ECRLS-MVSBs method. In addition, the computational time required by the SB-MVSBs decreases with the increase of the continuity order m. The reduction of the computational time is caused by the fact that the kernel space bases based transformation has cut down the scale of the original spline basis based model.

4

Recursive Sequential Multivariate Simplex B-splines on Aerodynamic Model Identification

The computational efficiency of an existing multivariate simplex B-splines (MVSB) based recursive identification method still needs to be improved. Although the substitution based MVSB (SB-MVSB) presented in Chapter 3 is more efficient than the ECRLS-MVSBs method, it might be better to develop a more efficient recursive identification approach for the MVSB regression framework in order to allow for online real-time model identification of a complex nonlinear system.

In this chapter, a local per-simplex model updating technique is developed to speed up a recursive MVSB method. Since this new method contains two sequential mathematical operations at each updating cycle, it is referred to as a recursive sequential MVSB method in the remainder of this thesis. In this approach, the overall input space is divided into multiple sub-domains. At each updating time step, only the covariance matrix which belongs to the currently active sub-domain is updated. In order to guarantee global smoothness of the identified model, a smoothing (smearing) step, i.e., the second sequential step, based on an optimal linear projection is performed at each updating time instant.

Avoiding high computational loads is essential to online aerodynamic model identification algorithms, which are at the heart of any model-based adaptive flight control system. Multivariate simplex B-spline (MVSB) methods are excellent function approximation tools for modeling the nonlinear aerodynamics of high performance aircraft. However, the computational efficiency of the MVSB method must be improved in order to enable real-time onboard applications, for example in adaptive nonlinear flight control systems.

In this chapter, a new recursive sequential identification strategy is proposed for the MVSB method aimed at increasing its computational efficiency, thereby allowing its use in onboard system identification applications. The main contribution of this new method is a significant reduction of computational load for large scale online identification problems as compared to the existing MVSB methods. The proposed method consists of two sequential steps for each time interval, and makes use of a decomposition of the global problem domain into a number of subdomains, called modules. In the first step the B-coefficients for each module are estimated using a least squares estimator. In the second step the local B-coefficients for each module are then smoothened into a single global B-coefficient vector using a linear minimum mean square errors (LMMSE) estimation. The new method is compared to existing batch and recursive MVSB methods in a numerical experiment in which an aerodynamic model is recursively identified based on data from an NASA F-16 wind-tunnel model.

4.1. INTRODUCTION

In the implementation of any model-based adaptive flight control system, like for example nonlinear dynamic inversion control [87] and modular adaptive backstepping control [47, 163], it is critical to at all times maintain an accurate onboard model of the aircraft under control. Central to such a control system is a recursive system identification loop, which constantly updates the onboard aerodynamic model as new measurements become available [102, 103]. It is essential that this loop is computationally efficient, as onboard computational capabilities in most applications are severely limited. In the past, the field of recursive system identification has been well studied, like for example variants of recursive least square methods and maximum likelihood method, see, e.g., [82, 133] and [83]. More recently, it has been suggested that recursive identification methods are developing towards the direction of nonlinear recursive identification methods [37, 51, 115, 120, 167, 176, 179].

The field of recursive identification can be split into parametric methods and nonparametric methods. However, this chapter will only focus on the parametric methods. A classical online parametric identification method is the ordinary polynomial basis based (OPBB) method. This method has been intensively studied by Klein, Morelli et al. [68, 69, 88, 100, 137]. The model structure selection problem is solved by ranking the polynomial terms according to their effect factors using the orthogonal least squares methods [68, 137].

A recent promising parametric identification method is the multivariate simplex B-splines (MVSB) method [9, 10, 34, 37, 73]. In [34] a new identification method was introduced which used the B-form basis polynomials of multivariate simplex B-splines in a linear regression framework. A batch estimation method is then adopted to identify the coefficients of the splines, also known as B-coefficients. This linear regression based MVSB scheme is a full-domain verifiable method due to the fact that its functional output are bounded by the maximum and the minimum B-coefficients [73]. In comparison with the OPBB identification method, the B-spline basis used in the MVSB method enjoys higher numerical stability since it is defined in terms of normalized Barycentric Coordinates [10, 73]. In addition, the approximation power of the MVSB method can be increased far beyond the capabilities of any OPBB method by

dividing the flight envelope into any number of subdomains called simplices [34, 73], which are organized in a structure called a triangulation. Continuity constraints are used to enforce a predefined continuity order between neighboring simplices in the triangulation, thereby ensuring that the resulting spline function is globally continuous [10, 73]. Compared with the well-known multivariate tensor product splines, the advantage of MVSB method is that it can fit scattered multidimensional datasets on non-rectangular domains [10, 20, 73].

Given the structure of the triangulation and the polynomial order of the basis functions, the MVSB identification problem can be formulated as a constrained linear system which can be solved using for example Lagrange multipliers [10]. The recursive identification scheme from [180] has been introduced into MVSB theory by de Visser et al. [37] in the form of the equality constrained recursive least squares (ECRLS) method. In the ECRLS method, the B-coefficients are recursively updated using the newly available data. This recursive identification algorithm has shown to be significantly more efficient in terms of computational load than the above mentioned batch method. However, real-time application of the MVSB is still hard to achieve due to the relatively high computational complexity when the B-coefficient vector contains more than one thousand elements. When applied to the problem of aerodynamic model identification of aircraft with highly nonlinear aerodynamics and extensive flight envelopes, the total number of elements in the B-coefficient vector can easily exceed this number. This may be either caused by a high resolution of the triangulation, a high polynomial order within each simplex, or a combination of both.

The objective of this chapter is to present an improved recursive identification method using MVSB that has a lower computational load than the ECRLS method, while having a comparable approximation power. It is found that the computational efficiency of the ECRLS method can be further improved by avoiding the updating of the global covariance matrix, which becomes highly time-consuming for large scale problems. The merit of the new method is that it avoids the computation of a global covariance matrix altogether, and therefore its computational efficiency can be higher than recursive MVSB methods like ECRLS that do require the updating of a global covariance matrix.

Simulated flight data, generated with a subsonic F-16 flight simulation based on a NASA wind-tunnel dataset [114], is used to validate the proposed recursive sequential MVSB (RS-MVSB) method. The proposed new method is more efficient in computational terms than the batch MVSB and ECRLS methods at the cost of a minor loss in approximation power. As a baseline comparison, the OPBB method is also implemented. Our final goal is to provide an efficient MVSB method that is able to provide a global nonlinear aerodynamic model for the advanced model based flight control systems of the next generation fighter aircraft.

This chapter is organized as follows. In Section 4.2, preliminaries are given on multivariate simplex B-splines. In Section 4.3, the recursive sequential identification scheme for the multivariate simplex B-splines is proposed. The analysis of computationally complexity is provided in Section 4.4. In Section 4.5, the method is compared with both the batch method and the ECRLS method for computational aspects. The new method is applied in the identification of an aerodynamic model for the F-16 in Section 4.6. Finally, the conclusions and remarks are given in Section 4.7.

4.2. PRELIMINARIES ON MULTIVARIATE SIMPLEX B-SPLINES

The basic principles for simplex splines are briefly introduced in this section. Firstly, subsection 4.2.1 introduces how to compute barycentric coordinates for a single simplex. Secondly, the triangulation techniques are presented in subsection 4.2.2. Subsection 4.2.3 describes the calculation of the simplex B-splines basis within a single simplex. Then, subsection 4.2.4 presents the methodology of constructing the global basis vector, i.e., a B-form vector. Thereafter, the equality constraints on the global B-coefficient vector are provided in subsection 4.2.5. Finally, the spline function space is defined.

4.2.1. SIMPLEX AND BARYCENTRIC COORDINATES

Let t be an n-simplex formed by the convex hull of its $n+1$ non-degenerate vertices $(v_0, v_1, ..., v_n) \subset \mathbb{R}^n$. The normalized barycentric coordinates of some evaluation point $\mathbf{x} \in \mathbb{R}^n$ with respect to simplex t are defined as

$$b(\mathbf{x}) \quad := \quad (b_0, b_1, ..., b_n) \in \mathbb{R}^{n+1}, \ \mathbf{x} \in \mathbb{R}^n \tag{4.1}$$

which follows from the following implicit relation:

$$\mathbf{x} = \sum_{i=0}^{n} b_i v_i, \ \sum_{i=0}^{n} b_i = 1 \tag{4.2}$$

4.2.2. TRIANGULATIONS OF SIMPLICES

The approximation power of the multivariate simplex spline is partly determined by the structure of the triangulation. A triangulation $\mathcal{T}$ is a special partitioning of a domain into a set of J non-overlapping simplices:

$$\mathcal{T} := \bigcup_{i=1}^{J} t_i, \ t_i \cap t_j \in \{\varnothing, \tilde{t}\}, \forall \ t_i, t_j \in \mathcal{T} \tag{4.3}$$

with the edge simplex $\tilde{t}$ a k-simplex with $0 \leq k \leq n-1$. High quality triangulations can be obtained using constrained Delaunay triangulation (CDT) methods, such as the 2-dimensional CDT method presented by Shewchuk [128].

4.2.3. BASIS FUNCTIONS OF THE SIMPLEX B-SPLINES

According to [73] and [37], the Bernstein basis polynomial $B_\kappa^d(b(\mathbf{x}))$ of degree d in terms of the barycentric coordinates $b(\mathbf{x}) = (b_0, b_1, ..., b_n)$ from Eq. 4.2 is defined as:

$$B_\kappa^d(b(\mathbf{x})) := \begin{cases} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} & , \mathbf{x} \in t \\ 0 & , \mathbf{x} \notin t \end{cases} \tag{4.4}$$

where $\kappa = (\kappa_0, \kappa_1, ..., \kappa_n) \in N^{n+1}$ is a *multi-index* with the following properties: $\kappa! = \kappa_0!\kappa_1!...\kappa_n!$ and $|\kappa| = \kappa_0 + \kappa_1 + ... + \kappa_n$. In Eq. 4.4 we use the notation $b^\kappa = b_0^{\kappa_0} b_1^{\kappa_1} ... b_n^{\kappa_n}$. Given that $|\kappa| = d$, the total number of valid permutations of the *multi-index* κ is:

$$\hat{d} = \frac{(d+n)!}{n!d!} \tag{4.5}$$

In [30], it was proved that any polynomial $p(b)$ of degree d on a simplex t can therefore be written as a linear combination of $\hat{d}$ basis polynomials in what is known as the B-form as follows:

$$p^t(b(\mathbf{x})) := \begin{cases} \sum_{|\kappa|=d} c_\kappa^t B_\kappa^d(b(\mathbf{x})) & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases} \tag{4.6}$$

with c_κ^t the B-coefficients which uniquely determines $p^t(b(\mathbf{x}))$, where the superscript 't' indicates that p is defined on the simplex 't'. The total number of basis function terms is equal to $\hat{d}$, which is the total number of valid permutations of κ.

4.2.4. VECTOR FORMULATIONS OF THE B-FORM

As introduced in [37], the vector formulation, according to Eq. 4.6, for a B-form polynomial $p(b(\mathbf{x}))$ in barycentric $\mathbb{R}^{n+1}$ has the following expression:

$$p^t(\mathbf{x}) := \begin{cases} \mathbf{B}_t^d(b(\mathbf{x})) \cdot \mathbf{c}^t & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases}, \tag{4.7}$$

with $b(\mathbf{x})$ the barycentric coordinates of the Cartesian $\mathbf{x}$. The row vector $\mathbf{B}_t^d(b(\mathbf{x}))$ in Eq. 4.7 is constructed from individual basis polynomials which are sorted lexicographically[37].

The simplex B-spline function $s_d^m(b(\mathbf{x}))$ of degree d and continuity order m, defined on a triangulation consisting of J simplices, is defined as follows:

$$s_d^m(\mathbf{x}) := \mathbf{B}^d(b(\mathbf{x})) \cdot \mathbf{c} \ \in \mathbb{R}, \tag{4.8}$$

with $\mathbf{B}^d(b(\mathbf{x}))$ the global vector of basis polynomials which has the following full expression:

$$\mathbf{B}^d(b(\mathbf{x})) := [\ \mathbf{B}_{t_1}^d(b(\mathbf{x})) \quad \mathbf{B}_{t_2}^d(b(\mathbf{x})) \quad \cdots \quad \mathbf{B}_{t_J}^d(b(\mathbf{x}))\ ] \in \mathbb{R}^{1 \times J \cdot \hat{d}} \tag{4.9}$$

Note that according to Eq. 4.7 we have $\mathbf{B}_{t_j}^d(b(\mathbf{x})) = 0$ for all evaluation locations $\mathbf{x}$ that are located outside of the triangle t_j. This results in that $\mathbf{B}^d$ is a sparse row vector.

The global vector of B-coefficients $\mathbf{c}$ in Eq. 4.8 has the following formulation:

$$\mathbf{c} := \begin{bmatrix} \mathbf{c}^{t_1\top} & \mathbf{c}^{t_2\top} & \cdots & \mathbf{c}^{t_J\top} \end{bmatrix}^\top \in \mathbb{R}^{J \cdot \hat{d} \times 1} \tag{4.10}$$

with each $\mathbf{c}^{t_j}$ a per-simplex vector of lexicographically sorted B-coefficients.

For a single observation on y we have:

$$y = \mathbf{B}^d(b(\mathbf{x}))\mathbf{c} + r \tag{4.11}$$

with r the residue. Then, for all the N observations, we have the following well-known formulation:

$$\mathbf{y} = \mathbf{X}(b(\mathbf{x}))\mathbf{c} + \epsilon \in \mathbb{R}^{N \times 1} \tag{4.12}$$

with $\mathbf{X}(b(\mathbf{x})) \in \mathbb{R}^{N \times J \cdot \hat{d}}$ a collection matrix of the row vector $\mathbf{B}^d$ from Eq. 4.9, and $\epsilon = [r_1, r_2, ...r_N]^T$ the residue vector. For writing convenience, $\mathbf{X}(b(\mathbf{x}))$ will be written as $\mathbf{X}$ in the remainder of this chapter.

4.2.5. Global Continuity Constraints

To keep the smoothness among all subdomains, the following equality constraints should be maintained during the calculation of the global B-coefficient vector $\mathbf{c}$:

$$\mathbf{H} \cdot \mathbf{c} = 0 \tag{4.13}$$

with $\mathbf{H} \in \mathbb{R}^{(E \cdot R) \times (J \cdot \hat{d})}$ the smoothness matrix [34][10], R is the number of continuity conditions per edge. E is the number of edges in the specified triangulation. If all the simplices' surfaces connect smoothly on the edges within the whole triangulation, we call the simplex splines globally continuous. Global continuity is determined by Eq. 4.13.

4.2.6. Spline Function Space and a Polynomial Function Space

In this chapter, we use a new type of definition of polynomial function space:

$$P_d(n) := \{p_k(\mathbf{x}) : p_k|_{\mathbf{x}} \in \mathbb{P}_k,\ \forall \mathbf{x} \in \mathbb{R}^n \ and \ \forall k \leq d\} \tag{4.14}$$

with $\mathbf{x}$ the input vector, $\mathbb{P}_k$ the space of polynomials of degree k.

We use the following definition of the spline space, which is a modified form of the definition given by Lai et al. in [73]:

$$S_d^m(n) := \{s_d^m(\mathbf{x}) \in C^m : s_d^m|_{\mathbf{x}} \in \mathbb{P}_d,\ \forall \mathbf{x} \in \mathbb{R}^n\} \tag{4.15}$$

with $\mathbb{P}_d$ the space of polynomials of degree d, and n the dimension of function inputs.

Note that, the former represents the ordinary polynomial function bases with the order up to d. For example, if we select $\mathbf{x} = \left[x,\ y\right]^T$, then $P_2(2) := c_1 + c_2 x + c_4 y + c_3 x^2 + c_6 xy + c_5 y^2$ with x and y two elements of $\mathbf{x}$.

4.3. Recursive Sequential Identification Method with Multivariate Simplex Spline

The computational load of the batch [34] and ECRLS [37] methods increases to the point of being impractical for use in online identification when the complexity and required update rate of a system increases beyond a certain point. The new recursive sequential least squares (RS-LS) method circumvents this problem by negating the need to update an increasingly large covariance matrix, thereby providing an effective online identification method for large scale systems of high complexity. In this section, the theory of the RS-LS identification method is presented.

4.3.1. Theoretical Development

Combining Eq. 4.12 with Eq. 4.13, the linear regression formulation of the multivariate simplex B-splines is derived as follows:

$$\mathbf{y} = \mathbf{X} \cdot \mathbf{c} + \epsilon \tag{4.16}$$

$$\text{s.t. } \mathbf{H} \cdot \mathbf{c} = 0 \tag{4.17}$$

Assume that the singular value decomposition of $\mathbf{H}$ is as follows:

$$\mathbf{H}_{n\times m} = \mathbf{V}_{n\times n}\begin{bmatrix}\sum_{r\times r} & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(m-r)}\end{bmatrix}\mathbf{U}^{\top}_{m\times m} \tag{4.18}$$

where $\sum = \text{diag}\begin{pmatrix}\sigma_1 & , & \cdots & \sigma_r\end{pmatrix}$ is the diagonal vector of eigenvalues , $\sigma_1 \geq \cdots \geq \sigma_r > 0$, and r is the rank of $\mathbf{H}$. $\mathbf{V} = \begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2\end{bmatrix}$ is an n_{th} order orthogonal matrix, $\mathbf{V}_1$ is a n by r matrix. $\mathbf{U} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}$ is a m_{th}-order orthogonal matrix, $\mathbf{U}_1$ is a m by r matrix. It can be seen from Eq. 4.17 that $\mathbf{c} \in null(\mathbf{H})$.

Let $\boldsymbol{J}(\mathbf{c})$ be a least squares cost function of the global B-coefficient vector $\mathbf{c} = \left[\mathbf{c}^{t_j}\right]_{j=1}^{J} \in \mathbb{R}^{J\cdot\hat{d}\times 1}$ as follows:

$$\boldsymbol{J}(\mathbf{c}) = \left(\mathbf{y}-\mathbf{X}\mathbf{c}\right)^{\top}\left(\mathbf{y}-\mathbf{X}\mathbf{c}\right) \tag{4.19}$$

where $\mathbf{X} \in \mathbb{R}^{N\times J\cdot\hat{d}}$ is a matrix of B-form regressors for N observations as derived in [37], and vector $\mathbf{y}$ contains all N observations. J is the number of simplices, and $\hat{d}$ is the length of the coefficients vector located on each simplex.

A definition of the optimal linear minimum mean square error (LMMSE) estimation from [98] is stated as follows:
Given a set of linearly independent vector bases $\mathbf{M} = \{\eta_1, \eta_2, ..., \eta_{n-1}, \eta_n\}$, we use a linear combination $\hat{\varepsilon} = \sum_{i=1}^{n} a_i\eta_i$ to approximate the unknown random variable vector ε. The fitting error is defined as:

$$\xi = \varepsilon - \hat{\varepsilon} = \varepsilon - \sum_{i=1}^{n} a_i\eta_i \tag{4.20}$$

The mean square fitting error is defined as:

$$\mathbf{P} = \left|\left|\varepsilon - \sum_{i=1}^{n} a_i\eta_i\right|\right|^2 \tag{4.21}$$

Our mission is to derive n constant parameters $a_1, a_2, ...a_n$ that make the mean square error have the minimum value.

One theorem from [172] and [76] also needs to be introduced: If the set of vector bases $\mathbf{M} = \{\eta_1, \eta_2, ...\eta_{n-1}, \eta_n\}$ are orthonormal bases, then the linear minimum mean square error (LMMSE) estimation of the random variable vector ε is given by

$$\hat{\varepsilon} = \sum_{i=1}^{n}\langle\varepsilon, \eta_i\rangle\eta_i = \sum_{i=1}^{n} a_i\eta_i \tag{4.22}$$

where $\langle\varepsilon, \eta_i\rangle$ is the inner product of ε and η_i.

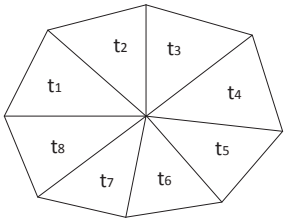
4.3.2. RECURSIVE SEQUENTIAL MULTIVARIATE SIMPLEX B-SPLINES

This method will be illustrated using a training dataset generated by the following function:

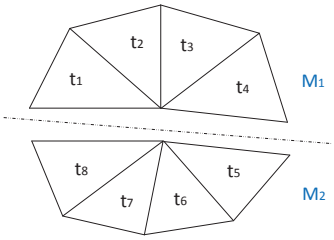
$$f = f(\mathbf{x}_1, \mathbf{x}_2) \tag{4.23}$$

where the column vectors $\mathbf{x}_1$, $\mathbf{x}_2$ are the inputs of function f. A spline function $s_d^m(b(\mathbf{x}_1,\mathbf{x}_2))$ is used to approximate function f.

A triangulation $\mathscr{T}$ is obtained using the Delaunay method according to the span in each dimension of $\mathbf{x} = [\mathbf{x}_1,\mathbf{x}_2]^T$. In the recursive sequential MVSB scheme, a modular subsystem is defined for each simplex (or set of simplices) of $\mathscr{T}$. Each modular subsystem defined on a small number of subdomains will be refered to as a module in the remainder of this chapter. An example of constructing modules among the whole triangulation for two dimensional training datasets is illustrated in Figure 4.1(a) and Figure 4.1(b). $s_1, s_2, ...s_8$ are eight simplices in $\mathscr{T}$ as shown in Figure 4.1(a). Two modules are constructed as shown in Figure 4.1(b), where each module contains four simplices. In practice, the number of simplices in each module is selected through a cross validation process.



(a) triangulation in two dimensions.



(b) module construction from triangulation.

Figure 4.1: Illustration of module construction.

Let $\mathbf{c}_i$ B-coefficient vector of the i^{th} module. In the recursive sequential method, we construct an independent estimation problem for each module. During every time step, we update the identification vector $\mathbf{c}_i$ which belongs completely to the specified module. Although updating the covariance matrix for each module is inevitable, the computation of the global covariance matrix is avoided. Because the global covariance matrix is many times larger than the local covariance matrix which belongs to one single module, this greatly reduces the computational time.

Let J be the number of simplices included in $\mathscr{T}$. The discontinuous global B-coefficient vector $\mathbf{c}$ has the following structure:

$$\mathbf{c} = \left[\mathbf{c}_1{}^\top \mathbf{c}_2{}^\top \ldots \mathbf{c}_i{}^\top \mathbf{c}_J{}^\top\right]^\top \tag{4.24}$$

The RS-MVSB method is comprised of two consecutive steps for each new data point: 1)updating the B-coefficient vector $\mathbf{c}_i$ of one single module and constructing vector $\mathbf{c}$; 2) deriving the smooth global B-coefficient vector $\tilde{\mathbf{c}}$ from the discontinuous vector $\mathbf{c}$ by some methodology under the constraints Eq. 4.17.

Given the splines' triangulation $\mathscr{T}$, the kernel space matrix $\mathbf{U}_2$ for the linear mapping transformation is fixed and thus the orthonormal basis vectors which are columns of $\mathbf{U}_2$ are fixed and time invariant. According to [96], the smooth vector $\tilde{\mathbf{c}}$ which satisfies Eq. 4.17 should have the same degree of freedom as the kernel space determined by the

columns of matrix $\mathbf{U}_2$. Inspired by this conclusion, a hard curtailing method is proposed to enforce the continuity constraints Eq. 4.17 at each recursive time step.

According to Eq. 4.17, vector $\widetilde{\mathbf{c}}$ should be located in the kernel space of H which is determined by $\mathbf{U}_2$ as shown in Eq. 4.18. In addition, the column vectors of matrix $\mathbf{U}_2$ are orthonormal bases of space $null(\mathbf{H})$ according to [96]. Let

$$\mathbf{U}_2 = \{\eta_1, \eta_2, ...\eta_{n-1}, \eta_n\} \tag{4.25}$$

By means of linear minimum mean square error (LMMSE) estimation from Eq. 4.22, the final smooth global B-coefficient vector $\widetilde{\mathbf{c}}$ is derived from the discontinuous global B-coefficient vector $\mathbf{c}$ as follows:

$$\widetilde{\mathbf{c}} = \sum_{i=1}^{n} \langle \mathbf{c}, \eta_i \rangle \eta_i \tag{4.26}$$

Figure 4.2 provides an overview of the overall adaptive control system. The function of the 'Onboard Spline Model' block is to reconstruct an aircraft aerodynamic model, and the 'Inner-loop Controller' block is aimed at producing part of control input signals using the identified aerodynamic coefficients. As can be seen from Figure 4.2, the recursive sequential setup can be summarized into three steps:

1) As a preparation, the $\mathbf{U}_2$ matrix is derived from Eq. 4.18, and a local module for each simplex of $\mathcal{T}$ triangulation is constructed. Let the number of simplices be J, then J estimation sub-problems are constructed.

2) When a new data point is available, the simplex index is used to decide which module it belongs to. Suppose that the current data point belongs to the i^{th} module, then updating the local parameter vector segment $\mathbf{c}_i$ of the i^{th} module using a least squares algorithm. Then we can get the corrected $\mathbf{c}$ according to Eq. 4.24. Simultaneously, the regressors' covariance matrix $\mathbf{P}_i$ of the i^{th} module is also updated.

3) The smooth global B-coefficient vector $\widetilde{\mathbf{c}}$ is derived from $\mathbf{c}$ using LMMSE estimation according to Eq. 4.26. Then, go to step 2 and wait for new data points.

It should be noted here that the first step of the recursive sequential method is independent of the second step. Therefore, we are allowed to skip the smoothing step (LMMSE estimation) occasionally as long as the smoothed vector $\widetilde{\mathbf{c}}$ is not readily needed in order to reduce the computational time. The main contribution of this algorithm is that updating the global covariance matrix is avoided in using recursive least square method, and therefore computational load is greatly reduced.

4.4. COMPUTATIONAL COMPLEXITY

The batch method, presented in [34], contains a matrix inversion. Therefore, the batch method has a computational complexity of $\mathcal{O}(m^3)$ with m the dimensionality of the covariance matrix if it uses the Gaussian elimination to calculate the matrix inversion. The computational complexity of the ECRLS method, presented in [180] and [37], is $\mathcal{O}(3m^2)$ [180] which is the result of the recursive updating of the global covariance

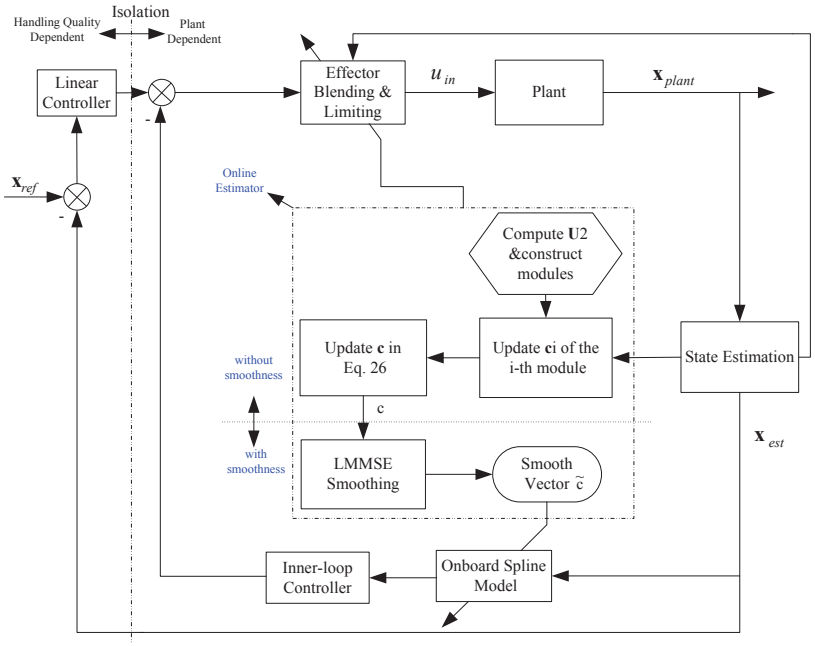


Figure 4.2: Structure of an adaptive control system based on multivariate spline model.(Computational flow chart of the RS-LS method is described in the central block.)

matrix. Therefore, the ECRLS method is more suitable for real-time applications because the number of algebraic operations and the required memory volumes are reduced in each cycle [180].

The computational complexity of the RS-MVSB method presented in Section 4.3 is the result of two operations. The first operation is the updating of the global covariance matrix at each cycle. Since the computational complexity of the ECRLS method is $\mathcal{O}\left(3m^2\right)$, we can calculate the computational time which is consumed in updating the covariance matrix of the RS-MVSB method by substituting the scale of the identification problem. Assume the number of modules in the $\mathcal{T}$ triangulation is k, then the computational complexity of updating the covariance matrix is $\mathcal{O}\left(3\left(\frac{m}{k}\right)^2\right) = \mathcal{O}\left(3\frac{m^2}{k^2}\right)$. The second operation is the smoothing of the global B-coefficient vector by Eq. 4.22:

$$\mathcal{O}\left(m\left(m-r\right)+m\left(m-r\right)\right) \approx \mathcal{O}\left(2m\left(m-r\right)\right) \tag{4.27}$$

where r is the rank of the smoothness matrix $\mathbf{H}$, and only the multiplication operation is considered.

As mentioned in Section 4.3.2, the smoothing step in the recursive sequential method is not necessary to be executed if the smooth global covariance matrix is not readily needed. In this case, the computational complexity of the overall method reduces to $\mathcal{O}\left(3\frac{m^2}{k^2}\right)$ in each cycle. The computational complexity (CC) analysis is concluded in

Table 4.1: Computational Complexity (CC) in time

Methods	batch	ECRLS	RS-LS(R)	RS-LS(N)
CC	$\mathcal{O}\left(m^3\right)$	$\mathcal{O}\left(3m^2\right)$	$\mathcal{O}\left(3\frac{m^2}{k^2}\right)$	$\mathcal{O}\left(2m^2-2mr+3\frac{m^2}{k^2}\right)$

Table 4.1.

In Table 4.3, the letter 'R' is an acronym of 'Reduced', which means the LMMSE smoothing process is carried out every 1000 time steps. Accordingly, the letter 'N' is an abbreviation of 'Not reduced'.

4.5. COMPUTATIONAL ASPECTS

In this section, the recursive sequential method is compared with the batch method presented in [34] and the ordinary recursive least squares (ECRLS) method presented in [37]. Firstly, the recursive sequential least squares based multivariate spline method is applied to fit a bivariate dataset, and compared with the batch method and ECRLS method. Secondly, it was validated by being applied to approximate a trivariate dataset in a hybrid manner using multi-function structure. Thirdly, the proposed method was applied to fit the same three dimensional dataset but with a different triangulation structure, where total number of B-coefficients is larger than 2000 as a result of the increased number of simplices. This experiment is designed for demonstrating the advantage in terms of computational efficiency of the recursive sequential MVSB (RS-MVSB) method when it is applied to large scale problems. All the numerical experiments, presented in this chapter, were implemented in the Matlab running as interpreted scripts and run on a quad core PC (Intel Xeon E3-1270@3.40 GHZ, RAM 16.0 GB).

4.5.1. DEMONSTRATION SETUP

In this section, two datasets are used. The first one is a trivariate dataset χ_{3d} which consists of 22000 scattered data points in $\mathbb{R}^3$, generated using a uniform random number generator in the interval $\{[0,1],[0,1],[0,1]\}$:

$$\chi_{3d} = \{x_1, x_2, x_3\} \in \{U[0,1], U[0,1], U[0,1]\} \tag{4.28}$$

The output measurements of the data points were generated using a three dimensional compound function:

$$f_t(x_1, x_2, x_3) = 0.33\left\{f_1(x_1, x_2, x_3) + f_2(x_1, x_2)\right\} + k_1 \cdot v \tag{4.29}$$

where $k_1 \cdot v$ is a uniformly distributed white noise sequence of magnitude k_1, the subfunctions take the following formulation:

$$f_1(x_1, x_2, x_3) = x_1 x_2 + x_3 \sin(1.5x_2 + x_1) \tag{4.30a}$$

$$f_2(x_1, x_2) = {x_2}^2 \sin(2x_1 + 10) + x_1 \cos(1.5x_2) \tag{4.30b}$$

20000 data points are used as training data and the remaining 2000 data are used as validation data.

Table 4.2: Comparison in 2-D, 168 parameters, 20000 data points

Methods	batch	ECRLS	RS-LS(R)
normalized time	514.1	1.0	0.6110
time per cycle(ms)	53.97	0.105	0.0641
max ($\|\epsilon\|$)	0.0792	0.0797	0.0839
RMSE (ϵ)	0.0201	0.0201	0.0201

The second dataset χ_{2d}, which contains 20000 data points in $\mathbb{R}^2$, is a bivariate dataset with its outputs generated using Eq. 4.30b together with $k_1 \cdot v$. In all of the following numerical experiments, we selected $k_1 = 0.02$.

4.5.2. COMPARISON WITH THE BATCH METHOD AND ECRLS METHOD

In the first numerical experiment, the bivariate training dataset χ_{2d} consisting of 20000 scattered data points is used. For the multivariate spline function $S(x) = S_d^m(n)$, we select n=2, d=5, m=1, where n,d, and m are defined in Eq. 4.15. The structure of the triangulation is determined by vertices located on the grid $\{[0, 0.5, 1], [0, 0.5, 1]\}$.

The recursive sequential least squares (RS-LS) method developed in Section 4.3 was compared with the batch method presented in [34] and the ordinary recursive least squares (ECRLS) method presented in [34]. The comparison results are shown in Table 4.2. The length of the B-coefficient vector $\tilde{\mathbf{c}}$ is 168. The total computational time, which is shown in the first row, is normalized using the computational time of ECRLS method. Apparently, the RS-LS method enjoys a significantly lower computational time than the batch method and ECRLS method while maintaining equivalent fitting accuracy (implied by max ($|\epsilon|$) and root mean squared error (RMSE)). The RMSEs are very close to the magnitude of the white noise add into the training data.

In the second numerical experiment, the RS-LS method is used to approximate the three dimensional dataset χ_{3d} consisting of 20000 data points, and the results are elaborated in Table 4.3. In this case, a compound spline function $S(x) = S_{d_1}^{m_1}(n_1) + S_{d_2}^{m_2}(n_2)$ consisting of a trivariate and a bivariate simplex B-spline function is used to approximate the function in Eq. 4.29. We selected $n_1 = 3$, $d_1 = 5$, $m_1 = 0$ and $n_2 = 2$, $d_2 = 4$, $m_2 = 1$. In the implementation of the recursive sequential method, a virtual covariance matrix with block diagonal form is constructed in Eq. 4.31.

$$\mathbf{C}_g = \begin{bmatrix} \mathbf{C}_{11} & & & & & & \\ & \mathbf{C}_{12} & & & & & \\ & & . & & & & \\ & & & \mathbf{C}_{1i} & & & \\ & & & & . & & \\ & & & & & \mathbf{C}_{1m} & \\ & & & & & & \mathbf{C}_{2t} \end{bmatrix} \tag{4.31}$$

where $\mathbf{C}_{1i}$ is covariance matrix of the i^{th} simplex in triangulation $\mathcal{T}_1$, $\mathbf{C}_{2t}$ is the global covariance matrix of the triangulation $\mathcal{T}_2$. Let n_1 and n_2 be the total number of B-coefficients which belong to $\mathcal{T}_1$ and $\mathcal{T}_2$ separately, then the size of the square matrix $\mathbf{C}_{1i}$ is $\frac{n_1}{m}$ with m the number of modules in $S_{d_1}^{m_1}(n_1)$ and the size of the square matrix $\mathbf{C}_{2t}$

Table 4.3: Comparison in 3-D, 366 parameters, 20000 data points

Methods	batch	ECRLS	RS-LS(R)	RS-LS(N)
normalized time	709.0420	1.0	0.3150	0.4057
per cycle(ms)	348.9557	0.4922	0.1552	0.1997
max ($\|\epsilon\|$)	0.0795	0.0750	0.0745	0.0736
RMSE (ϵ)	0.0200	0.0199	0.0205	0.0201

Table 4.4: Comparison in 3-D, 2808 parameters, 20000 data points

Methods	batch	ECRLS	RS-LS(R)	RS-LS(N)
normalized time	243.42	1.0	0.0154	0.1317
per cycle(ms)	10094.4	41.4682	0.6390	5.4641
max ($\|\epsilon\|$)	0.0824	0.0792	0.0860	0.0915
RMSE (ϵ)	0.0200	0.0201	0.0253	0.0242

is n_2. Assume the current data point is located in the i^{th} module of $S_{d_1}^{m_1}(n_1)$, then the local covariance matrix for the current time step is:

$$\mathbf{C}_l = \begin{bmatrix} \mathbf{C}_{1i} & \\ & \mathbf{C}_2 \end{bmatrix} \tag{4.32}$$

To get the results in Table 4.3, the support vertices $\{[0,1],[0,1],[0,1]\}$ were selected to construct the $\mathcal{T}$ triangulation. The length of the B-coefficient vector $\tilde{\mathbf{c}}$ is 366.

It can be seen from Table 4.3 that both the ECRLS method and the RS-LS method are significantly faster than the batch method. The best performing RS-LS method, i.e., RS-LS(R-s), is more than 2000 times faster than the batch method. According to column 2 and 3, RS-LS method is 3 times faster than the ECRLS method while maintaining nearly the same approximation ability (implied by the RMSE). Note that, the RMSEs are very close to the magnitude of the white noise added to the training data.

Thirdly, the advantage of the RS-MVSB method for large scale and multi-function identification problems is demonstrated using the three dimensional dataset χ_{3d}, which contains 20000 data points as described in Section 4.5.1. In this numerical experiment, a spline function $S(x) = S_{d_1}^{m_1}(n_1) + S_{d_2}^{m_2}(n_2)$ is used to approximate the function $f_t(x_1, x_2, x_3)$ shown in Eq. 4.29. A different triangulation $\mathcal{T}$ was constructed using the support vertices $\{[0, 0.5, 1], [0, 0.5, 1], [0, 0.5, 1]\}$. The B-spline function $S_{d_1}^{m_1}(n_1)$ was constructed using $n_1 = 3$, $d_1 = 5$, $m_1 = 0$, and $S_{d_2}^{m_2}(n_2)$ was constructed using $n_2 = 2$, $d_2 = 4$, $m_2 = 1$. The length of the global B-coefficient vector $\tilde{\mathbf{c}}$ is 2808.

In this numerical experiment, 20000 data points from dataset χ_{3d} from Eq. 4.28 were used as training data for the batch method, the ECRLS method and the RS-LS method. The results in Table 4.4 clearly show that the ECRLS method is more than 200 times faster than the batch method, while the RS-LS method is 60 times faster than the ECRLS method while the approximation power (implied by RMSE) is close to that of the batch method. The RMSE approaches the magnitude of the white noise added to the training data. Again, this speedup is the direct result of not having to update a global covariance matrix.

4.6. Application on Aerodynamic Model Identification of the F-16

In this section, the recursive sequential algorithm was applied to identify an aerodynamic model of the F-16 fighter aircraft based on data from a NASA wind-tunnel database of the F-16 [114]. The next generation of fighter aircraft will have highly nonlinear aerodynamics. In order to obtain accurate aerodynamic models for such aircraft, their full flight envelopes need to be partitioned using high resolution triangulations. The result of this is a longer B-coefficient vector for the simplex B-splines. In case the aerodynamic model needs to be updated online, for example after a damage event, the batch and ECRLS methods may become infeasible due to their low computational efficiency. This section is aimed at demonstrating the approximation power of the new RS-MVSB method when dealing with a large scale, complicated nonlinear modeling real-world problem. The RS-MVSB method is compared with both the polynomial basis based identification method and the batch MVSB method. In the remainder of this chapter, we assume that all aircraft states have been obtained from the 'State Estimation' block, e.g., an extended Kalman filter, shown in Figure 4.2. The model identification method proposed in this chapter forms the second step of the so-called two step method [108].

4.6.1. F-16 Aerodynamic Model Structure

$$\begin{aligned} F_x\left(\alpha,\beta,\delta_e,\delta_{lef},\frac{q\bar{c}}{V}\right) &= f_1(\alpha,\beta,\delta_e)+f_2(\alpha,\beta)\cdot\delta_{lef} \\ &+ f_3(\alpha)\cdot\frac{q\bar{c}}{V}+f_4(\alpha)\cdot\frac{q\bar{c}}{V}\cdot\delta_{lef} \end{aligned} \tag{4.33}$$

According to Eq. 4.33, we can derive the following linear regression formulation for compound multivariate B-splines.

$$F_x = F_{x1}F_{x2} \tag{4.34}$$

where

- $F_{x1}=\left[\mathbf{B}_1,\mathbf{B}_2\cdot\delta_{lef},\mathbf{B}_3\cdot\frac{q\bar{c}}{V},\mathbf{B}_4\cdot\frac{q\bar{c}}{V}\cdot\delta_{lef}\right]$
- $F_{x2}=\left[\mathbf{c}_1^\top,\mathbf{c}_2^\top,\mathbf{c}_3^\top,\mathbf{c}_4^\top\right]^\top$

For simplicity, let $\mathbf{B}_3=\mathbf{B}_4$, then the third term and the fourth term in Eq. 4.34 can be merged:

$$F_x = F_{x3}F_{x4} \tag{4.35}$$

where

- $F_{x3}=\left[\mathbf{B}_1,\mathbf{B}_2\cdot\delta_{lef},\mathbf{B}_3\cdot\frac{q\bar{c}}{V}\cdot(1+\delta_{lef})\right]$
- $F_{x4}=\left[\mathbf{c}_1^\top,\mathbf{c}_2^\top,\mathbf{c}_3^\top\right]^\top$

According to de Visser [35], the global continuity matrix for the combined spline function has the following expression:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & & \\ & \mathbf{H}_2 & \\ & & \mathbf{H}_3 \end{bmatrix} \tag{4.36}$$

with $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3$ the continuity matrices of the three independent sub functions.

In the implementation of the RS-MVSB method, a virtual covariance matrix in the block diagonal form is constructed in Eq. 4.37.

$$\mathbf{C}_g = \begin{bmatrix} \mathbf{C}_{tn_1} & & & & & & \\ & \mathbf{C}_{tn_2} & & & & & \\ & & \cdot & & & & \\ & & & \mathbf{C}_{tn_i} & & & \\ & & & & \cdot & & \\ & & & & & \mathbf{C}_{tn_m} & \\ & & & & & & \mathbf{C}_{t_{n2}} \\ & & & & & & & \mathbf{C}_{t_{n3}} \end{bmatrix} \tag{4.37}$$

where $\mathbf{C}_{tn_i}$ is the covariance matrix of the i^{th} simplex in triangulation $\mathcal{T}_1$, $\mathbf{C}_{n2}$ and $\mathbf{C}_{n3}$ are covariance matrices of triangulation $\mathcal{T}_2$ and $\mathcal{T}_3$ separately. In this equation, $tn_1 + tn_2 + ... + tn_i + ... + tn_m = n1$ where $n1$, $n2$ and $n3$ are the total number of B-coefficients which belong to $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$ separately. The local covariance matrix at the current time step is:

$$\mathbf{C}_l = \begin{bmatrix} \mathbf{C}_{tn_i} & & \\ & \mathbf{C}_{t_{n2}} & \\ & & \mathbf{C}_{t_{n3}} \end{bmatrix} \tag{4.38}$$

4.6.2. ORIGINAL WIND TUNNEL DATA FOR AN F-16 AIRCRAFT

The inputs of the training dataset were obtained by randomly generating 20000 input states inside the domain of the NASA wind-tunnel model. The inputs of the validation dataset containing 4331 points are produced by the grids determined by α and β. The outputs of both datasets were calculated using Eq. 4.33 based on the wind-tunnel data provided by [114] with $\delta_{lef} = 1\,degree, V = 600\,ft/s, q = 0.1\,rad/s, \bar{c} = 11.32\,m$. To investigate the effects from the measurement noise, the white noise with a magnitude of 1% (relative value to the corresponding upper and lower extreme) has been added to the model outputs, i.e., C_X, C_m, in generating the training datasets. This observation noise together with Eq. 4.33 is playing the same role as Eq. 4.29.

In this chapter, we are only concerned with the identification of the longitudinal force and moment coefficients. In addition, only the results for C_X and C_m are provided since they show much more significant nonlinearities than C_Z. Three-dimensional surfaces of longitudinal force coefficient C_X and moment coefficient C_m when $\delta_e = 0$ were plotted in Figure 4.3 and Figure 4.4 separately. Highly nonlinear behavior of the aerodynamic coefficients can be observed in these figures.

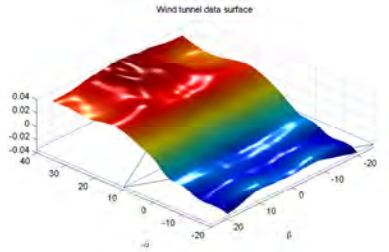


Figure 4.3: C_X wind-tunnel data surface ($\delta_e = 0^o$, $\delta_{lef} = 1^o$, $\frac{q\bar{c}}{V} = 0.0018866$).

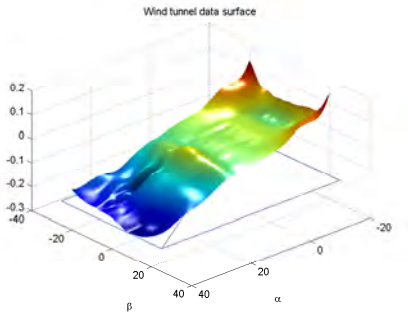


Figure 4.4: C_m wind-tunnel data surface ($\delta_e = 0^o$, $\delta_{lef} = 1^o$, $\frac{q\bar{c}}{V} = 0.0018866$).

4.6.3. MODEL STRUCTURE SELECTION AND RECURSIVE SEQUENTIAL IDENTIFICATION

In the simulation, the MVSB function approximator contains three subfunctions: $S(x) = S_{d_1}^{m_1}(n_1) + S_{d_2}^{m_2}(n_2) + S_{d_3}^{m_3}(n_3)$, where we selected $n_1 = 3$, $n_2 = 2$, $d_2 = 4$, $m_2 = 1$; $n_3 = 1$, $d_3 = 3$, $m_3 = 1$ with d_1, m_1 undetermined. That is, the compound spline function is the following: $S(x) = S_d^m(3) + S_4^1(2) + S_3^1(1)$. The partitioning vector of α is $\begin{bmatrix}-20 & 10 & 40\end{bmatrix}$. The partitioning vector of β is $\begin{bmatrix}-25 & 25\end{bmatrix}$. The partitioning vector of δ_e is $\begin{bmatrix}-20 & 20\end{bmatrix}$. The inputs were normalized into the range of $[0\,1]$ in order to further improve the approximation accuracy. In order to select a suitable model structure for the spline model, the effects of the structural parameters, i.e., d and m, will be investigated.

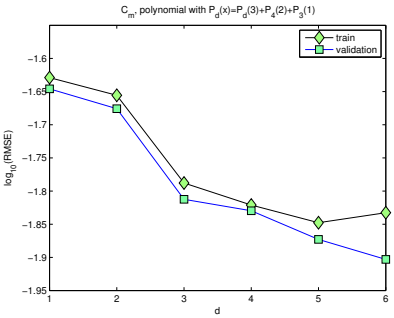
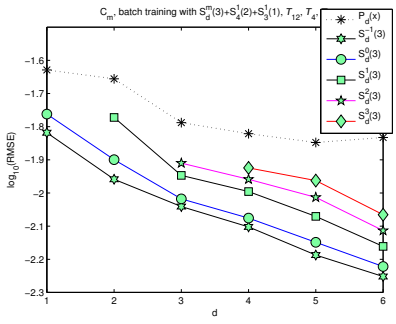
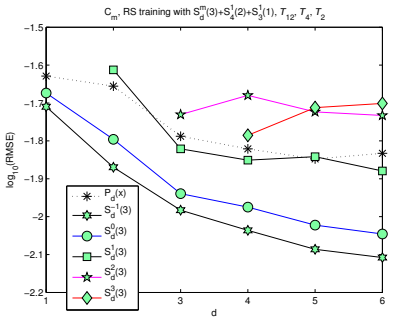


Figure 4.5: Different combination of d for $P_d(3)$ together with $P_4(2)$ and $P_3(1)$.

The results of the identification experiment with the OPBB method are shown in Figure 4.5. The approximation results on the training dataset using batch MVSB method and RS-MVSB method are given separately in Figure 4.6. Figure 4.6(a) and Figure 4.6(b) clearly shows that the approximation accuracy of both methods will increase with the increase of d, while decrease with the increase of continuity order m. Apparently, the approximation accuracy of the RS-MVSB method is very close to that of the batch MVSB

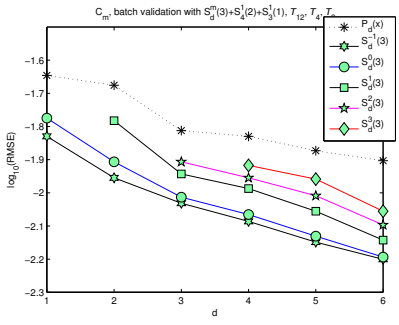


(a) Batch training.

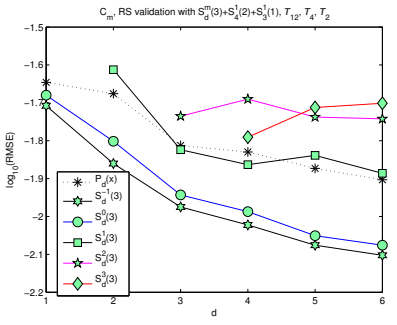


(b) RS-MVSB training.

Figure 4.6: Different combination of m and d for $S_d^m(3)$ together with $S_4^1(2)$ and $S_3^1(1)$.



(a) Batch test.



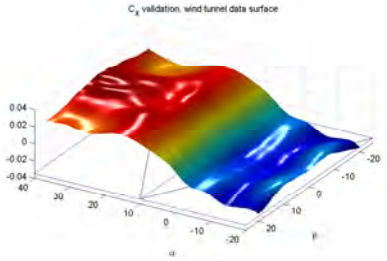
(b) RS-MVSB test.

Figure 4.7: Different combination of m and d for $S(3)_d^m$ together with $S_4^1(2)$ and $S_3^1(1)$.

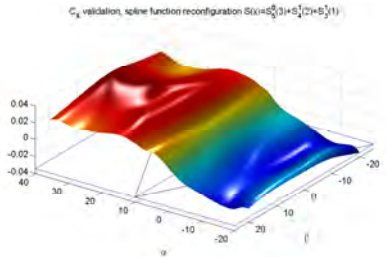
method when continuity order is less than two, although the performance of the new method is lower than the batch MVSB method when continuity order is high. Compared with Figure 4.5, Figure 4.6 also shows that the MVSB method has a higher approximation power than the OPBB method, especially when the polynomial order is larger than 3.

The approximation accuracy of the aforementioned methods are validated using 4331 validation data points; the RMSEs are plotted in Figure 4.7(a) and Figure 4.7(b) separately. According to Figure 4.7(a), the fitting accuracy of the Batch MVSB will increase with the increase of d and it will decrease with the increase of m. According to Figure 4.7(b), when the continuity order m is lower than 3, the fitting accuracy of the RS-MVSB will also increase when the polynomial order increases or the continuity order decreases. On the other hand, when the continuity order is equal to 3, the quality of fit will decrease with the increase of d. Comparing Figure 4.7(b) with Figure 4.7(a), it can also be seen that the RS-MVSB method has a slightly lower fitting accuracy than batch MVSB, which is the price paid for the increased computational efficiency of updating the B-coefficients.

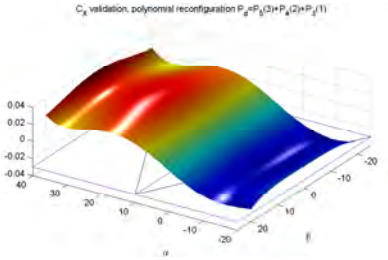
Finally, the performance of the RS-MVSB method is illustrated by comparing the predicted output on the 4331 knots with wind-tunnel data output. For the plotting of the 3 dimensional surface, we selected $\delta_e = 2^o$ in order to fix the third dimension δ_e. The spline function has the following structure: $S(x) = S_5^0(3) + S_4^1(2) + S_3^1(1)$. The structure of the OPBB approximator has the following form:$P(x) = P_5(3) + P_4(2) + P_3(1)$.



(a) Original wind-tunnel data surface.



(b) Spline function surface, $\mathcal{T}_{12}, \mathcal{T}_4, \mathcal{T}_2$.

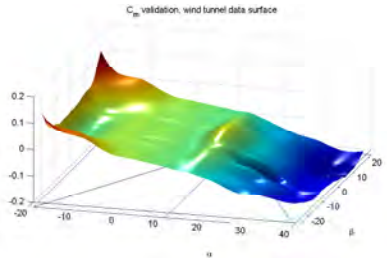


(c) Polynomial fitting surface.

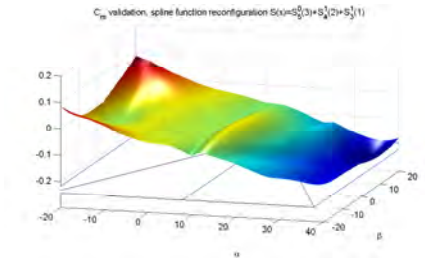
Figure 4.8: Validation surface of $C_X(\alpha, \beta, \delta_e = 2^o)$ with $\delta_{lef} = 1^o$, $\frac{q\bar{c}}{V} = 0.0018866$, 4331 data.

The wind-tunnel model for C_X for $\delta_e = 2^o$ is shown in Figure 4.8(a) as a comparison baseline. The spline function surface reconstructed from the estimated global B-coefficient vector $\tilde{\mathbf{c}}$ is given by Figure 4.8(b). Comparing Figure 4.8(b) and Figure 4.8(c) with Figure 4.8(a), it can be seen that the MVSB model closer resembles the wind-tunnel model than the OPBB model. The surfaces of C_m plotted in Figure 4.9(a), Figure 4.9(b), and Figure 4.9(c) show a similar result.

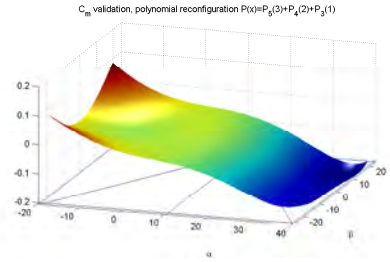
Finally, the most important advantage of the RS-MVSB method over the ECRLS MVSB method is that it requires a lower computational time at each recursive step when updating the B-coefficients. In the above mentioned simulation experiment, the length of the B-coefficient vector is 740 and the computational time of the RS-MVSB method is 0.9766 ms per data point. Whereas, the computational time of the ECRLS MVSB method is 3.6582 ms per data point. The low computational time requirement makes the RS-MVSB method more suitable for application in real-time adaptive flight control systems than the ECRLS MVSB method and the batch MVSB method.



(a) Original wind-tunnel data surface.



(b) Spline function surface, $\mathcal{T}_{12}, \mathcal{T}_4, \mathcal{T}_2$.



(c) Polynomial fitting surface.

Figure 4.9: Validation surface of $C_m(\alpha, \beta, \delta_e = 2^o)$ with $\delta_{lef} = 1^o$, $\frac{q\bar{c}}{V} = 0.0018866$, 4331 data.

4.7. CONCLUSIONS

A new recursive sequential strategy is developed and combined with the multivariate simplex B-splines with the aim of increasing the computational efficiency over existing multivariate spline based identification methods. This new recursive sequential multivariate B-splines method, called the RS-MVSB method, enables the online identification of large scale nonlinear aerodynamic models which are at the heart of modern adaptive model-based flight control systems. The RS-MVSB method consists of two sequential procedures at each time step. The first is the estimation of the B-coefficients for the local spline functions defined on the local simplex modules. The second procedure is the LMMSE-based smoothing step which results in a smooth global spline function.

The performance of the RS-MVSB method was compared to existing batch and recursive methods for simplex B-spline identification in a set of three numerical experiments. It was shown that the RS-MVSB method enjoys a significant advantage in terms of computational efficiency over the existing ECRLS method for large scale problems at a cost of a slight decrease in approximation power. In particular, the approximation power of the RS-MVSB method is comparable to the batch MVSB method and the ECRLS MVSB methods when the continuity order of the spline functions is lower than 2. For continuity orders larger than or equal to 2, the approximation power of the RS-MVSB method is lower than existing MVSB methods. In addition, all MVSB methods have a higher approximation power than the OPBB approximation method.

The RS-MVSB method is applied in the recursive identification of an aerodynamic

model using data from a NASA nonlinear wind-tunnel data set of the F-16 fighter aircraft. A comparison with the ordinary polynomial method and the batch MVSB method shows that the new RS-MVSB method has all the advantages of the MVSB method, i.e., high approximation power, while at the same time greatly reducing the required computational time in updating the spline model parameters.

5

ONLINE ADAPTIVE KERNEL METHOD BASED ON SUPPORT VECTOR REGRESSION

In Chapter 3 and Chapter 4, two different recursive identification methods using multivariate simplex B-splines (MVSB) were presented. Each of them has a higher computation efficiency than a regular equality constrained recursive least square (ECRLS) MVSB method. However, our research scope has so far been limited to the spline-based parametric method. Kernel methods, on the other hand, could also be an efficient model identification approach, and very well provide an accurate model approximation of a complex nonlinear system. Kernel methods, which can be parametric methods in a few cases, have received much attention from the industry in the past few decades and have shown their power in many real-world applications. Therefore, we explore whether there could be certain recursive kernel methods which are suitable for deriving a highly accurate global online aerodynamic model. In this chapter, a new recursive parametric kernel method is developed. Firstly, a support vector selection technique from the field of support vector machines is applied to determine kernel centres for a regular radial basis function (RBF) based kernel method. Secondly, we investigate how to modify or expand the kernel functions to enhance overall modelling accuracy by taking into account local data trends.

The optimality of the kernel number and kernel centers plays a significant role in determining the approximation power of nearly all kernel methods. However, the process of choosing optimal kernels is always formulated as a global optimization task, which is hard to accomplish. Recently, an improved algorithm called recursive reduced least squares support vector regression (IRR-LSSVR) was proposed for establishing a global nonparametric offline model. IRR-LSSVR demonstrates a significant advantage in choosing representing support vectors compared others. Inspired by the IRR-LSSVR, a new online adaptive *parametric kernel method called Weights Varying Least Squares*

Support Vector Regression (WV-LSSVR) is proposed in this chapter using the same type of kernels and the same centers as those used in the IRR-LSSVR. Furthermore, inspired by the multikernel semiparametric support vector regression, the effect of the kernel extension is investigated in a recursive regression framework, and a recursive kernel method called Gaussian Process Kernel Least Squares Support Vector Regression (GPK-LSSVR) is proposed using a compound kernel type which is recommended for Gaussian process regression. Numerical experiments on benchmark data sets confirm the validity and effectiveness of the presented algorithms. The WV-LSSVR algorithm shows higher approximation accuracy than the recursive parametric kernel method using the centers calculated by the k-means clustering approach. The extended recursive kernel method, i.e., GPK-LSSVR, has not shown any advantage in terms of global approximation accuracy when validating the test data set without real-time updates, but it can increase modeling accuracy if real-time identification is involved.

5.1. INTRODUCTION

Seeking optimality in selecting the number and the centers' position of the kernel basis functions has been a defining feature of applying the kernel basis function based identification method [126][28], particularly those dealing with complex dynamic systems. Given a modeling data set, the task of determining the centers and the number of the kernel basis functions becomes a global optimization problem, which usually requires a high computational load especially when the modeling data set is of large scale [21, 28]. Many contributions concerning kernel selection exist in the literature. Among many examples are the orthogonal least squares [26], clustering algorithms such as k-means [28], and more recently, many efforts in reducing the dependency between the kernel basis number and the training data number[21, 65, 120]. Despite the abovementioned contributions in selecting the centers and number of the kernel basis functions, a novel type of learning machine called support vector machine (SVM) has proven to be a good alternative. As suggested by Bernhard et al. [13], the SVM has an advantage in providing high-quality centers for other kernel methods. In other words, the kernel methods using the centers calculated by SVM enjoy higher approximation accuracy than the classical radial basis function (RBF) machine.

Support vector machines (SVMs) proposed based on structural risk minimization (SRM) principle are state of the art learning algorithms for pattern recognition and function approximation [28, 126, 165]. Specifically, it has been successfully applied in many fields, including feature selection [77], face recognition [79], image segmentation [169], and text classification [97]. To reduce the computational complexity in the training process, one branch of SVMs, namely least squares support vector machine (LSSVM), was proposed by Suykens et al. [145, 148]. In LSSVM, only a linear equation set needs to be solved rather than a quadratic programming problem as in classical SVMs. In terms of real-world application, LSSVM has achieved promising results in many different fields such as particle swarm optimization [80], fault diagnosis [7], thunderstorm prediction [123], foreign exchange rate forecasting [81], breast cancer diagnosis [121], aero-engine model reconstruction [177], and time series prediction [124]. However, LSSVM has also introduced a new problem, that is, its solution suffers from lack of sparseness and robustness [147]. In LSSVM, all training samples become support vectors owing

to the equality constraints, and its loss function takes a squared errors form. As a consequence, it needs a high computational load in training and lacks robustness. To overcome these drawbacks, many efforts have been made by Suykens [146], De Kruif and De Vries [32], Hoegaerts [56], Zeng and Chen [170] and Jiao [62] et al. For large data sets, Brabanter et al. [31] and Karsmakers et al.[65] recently developed the fixed-size kernel (SVR) modeling method. More recently, a novel and much sparser LSSVR method named improved recursive reduced LSSVR (improved recursive reduced least squares support vector regression (IRR-LSSVR)) is proposed by Zhao and Sun et al.[179] after combining a reduced technique [24] with the iterative strategy [62]. In each iteration, all the remaining non-support data points will be evaluated, and the data leading to a maximum reduction in the overall squared fitting errors is introduced. Above all, IRR-LSSVR exploits an improved criterion for selecting optimal support vectors recursively, which takes into account the adaptation of existing weights by the will-selected support vectors when introducing a new support vector. Consequently, IRR-LSSVR leads to a sparser SVM model when compared to other LSSVR methods. That is to say, it needs less support vectors while keeping almost the same approximation accuracy without greatly increasing the computational time in training [179].

Furthermore, the choice of kernel function type plays a paramount role in determining the modeling performance of a kernel method. When the modeling system has different data trends in different subdomains, the kernel method using single kernel commonly cannot lead to a satisfying result, i.e., the model does not globally fit the data. Multikernel learning algorithms [15, 74, 117, 178] have been intensively investigated during the last decade. Their superiority of leading to high approximation accuracy has been demonstrated by Ong et al.[117].

The objective of this chapter is to present an adaptive fixed-size kernel method called WV-LSSVR for real-time global model identification applications, which takes advantage of IRR-LSSVR in selecting kernel centers, for the global model real-time identification. Additionally, the multikernel effect on the kernel basis function based modeling method is also investigated, and a new online GPK-LSSVR kernel method is developed by extending the single Gaussian kernel into a compound kernel recommended for Gaussian process regression. GPK-LSSVR should be seen as an improved version of WV-LSSVR, which improves the approximation power of the WV-LSSVR method.

This chapter is structured as follows. In Section 5.2, the preliminaries on reduced LSSVR are briefly introduced. The new adaptive methods will be developed in Section 5.3. Results and analysis will be given in Section 5.4. Finally, the paper is concluded by Section 5.5.

5.2. PRELIMINARIES ON RECURSIVE REDUCED LSSVR

Because the centers of the kernels used in the laterly proposed kernel method will be calculated using recursive reduced LSSVR (RR-LSSVR), the preliminaries on RR-LSSVR are briefly introduced in this section.

5.2.1. REDUCED LEAST SQUARES SUPPORT VECTOR REGRESSION

Given a training data set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^m$ is the input with m-dimension and $d_i \in \mathbb{R}$ is its corresponding model output. The least squares support vector regression is solved by solving the following optimization problem:

$$\min_{w,e,b} \left\{ \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{N} e_i^2 \right\} \tag{5.1}$$

s.t. $d_i = w^T \varphi(x_i) + b + e_i$, $i = 1, ..., N$. where w is the normal vector of the hyperplane, b is the offset, $e = [e_1, ..., e_N]^T$ denotes the prediction residual vector, $C \in \mathbb{R}^+$ is the regularization parameter, $\varphi(\cdot)$ is the mapping from the input space to the feature space. In literature, this constrained optimization problem is solved by introducing Lagrangian factors:

$$L(w, b, e, \alpha) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{N} \alpha_i \left(d_i - w^T \varphi(x_i) - b - e_i \right) \tag{5.2}$$

5

where α is the Lagrangian multiplier vector. The conditions for optimality are

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i), \tag{5.3a}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0, \tag{5.3b}$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = C e_i, \tag{5.3c}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i - d_i = 0, \tag{5.3d}$$

Eliminating the vectors w and e, the following linear equations set is obtained:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \bar{K} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ d \end{bmatrix} \tag{5.4}$$

where $\mathbf{1} = [1_1, ..., 1_N]^T$, $d = [d_1, ..., d_N]^T$, $\bar{K}_{ij} = k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) + \delta_{ij}/C$ with

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}, i, j = 1, ..., N$$

$k(x_i, x_j)$ is the kernel function on the paired input vectors $\{(x_i, x_j), i = 1, ..., N; j = 1, ..., N\}$. The commonly used kernel function is the Gaussian defined by $k(x_i, x_j) = exp\left(-\|x_i - x_j\|^2 / 2\gamma^2\right)$. After obtaining the solution α from Eq. 5.4, for any new testing sample $x \in \mathbb{R}^m$, the predicting value is derived as follows:

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x) + b \tag{5.5}$$

Assuming that $\{(\mathbf{x}_i, d_i)\}_{i=1}^{M}$ is a reduced subset of original training samples, and are used to construct an approximation to the aforementioned normal LSSVR. From Eq. 5.3, the sparse expression of w can be rewritten as $w = \sum_{i \in S} \alpha_i \varphi(x_i)$ with S the index subset of $\{1, ..., N\}$. After substituting w into Eq. 5.1, we get the equivalent formulation of the optimization problem[176]:

$$\min_{b,\alpha_S} \left\{ L(b\alpha_S) = \frac{1}{2}\alpha_S^T K \alpha_S + \frac{C}{2}\sum_{i=1}^{N}\left(d_i - \sum_{j \in S}\alpha_j \cdot \varphi(x_j)^T \varphi(x_i) - b \right)^2 \right\} \tag{5.6}$$

where $K_{ij} = k(x_i, x_j)$, $i, j \in S$, α_S denotes the sub-vector of α indexed by S. Eq. 5.6 can be reformulated into the following economy form[176]:

$$\min_{b,\alpha_S} \left\{ L = [b\alpha_S^T] \left(\begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & K/C \end{bmatrix} + \begin{bmatrix} \mathbf{1}^T \hat{K} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \hat{K}^T \end{bmatrix} \right) \begin{bmatrix} b \\ \alpha_S \end{bmatrix} - 2\left([\mathbf{1}^T \hat{K}]\mathbf{d} \right)^T \begin{bmatrix} b \\ \alpha_S \end{bmatrix} \right\} \tag{5.7}$$

where $\hat{K}_{ij} = k(x_i, x_j)$, $i, j \in S$, $\mathbf{1}$ is a vector of all ones in an appropriate dimension, $\mathbf{0}$ is a vector of all zeros in an appropriate dimension. Let $\partial L/\partial b = 0$ and $\partial L/\partial \alpha_S = 0$, Eq. 5.7 can reach its optimal value at

$$\left(R + ZZ^T\right) \begin{bmatrix} b \\ \alpha_S \end{bmatrix} = Z\mathbf{d} \tag{5.8}$$

where

$$R = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & K/C \end{bmatrix}, \; Z = \begin{bmatrix} \mathbf{1}^T \\ \hat{K} \end{bmatrix}$$

By solving Eq. 5.8, a reduced LSSVR estimator is derived for a testing sample x:

$$f(x) = \sum_{i \in S} \alpha_i k(x_i, x) + b \tag{5.9}$$

5.2.2. IMPROVED ALGORITHM FOR SELECTING SUPPORT VECTORS

In IRR-LSSVR [179], an improved criterion was developed for RR-LSSVR. In each iteration, all the remaining non-support data points will be evaluated, and the data leading to a maximum reduction in the overall squared fitting errors is introduced. Above all, the adaptation of the weights by the will-selected support vectors is taken into account when introducing a new support vector.

5.3. PARAMETRIC KERNEL METHODS WITH THEIR CENTERS DETERMINED BY LSSVR

The property of a kernel, a nonlinear mapping function from the original space to the feature space, directly determines the approximation accuracy of SVM. Meanwhile,

sparseness plays a crucial role in making the kernel models enjoy a better prediction performance in terms of generalization performance and testing time. However, the kernel center selection process usually has a high computational time requirement. Since IRR-LSSVR has a superiority in efficiently selecting optimal locations and numbers of all kernels, it provides a tool for all the general kernel methods to determine the number and locations of the kernels when constructing the global model. In both kernel methods proposed in this chapter, IRR-LSSVR is utilized to deal with the pre-collected static training data pool to get the optimal support vectors $S = \{x_1, x_2, ..., x_M\}$, which are later used as kernel centers.

In many circumstances, like for example in real-time model identification, an online global kernel method provides a good alternative to the other possible online identification algorithms. In the remainder of this chapter, we will focus on developing an online kernel method based on IRR-LSSVR. A fixed-size online parametric kernel method has been developed by introducing a classical weights updating technique after the kernel centers have been determined using IRR-LSSVR.

As a further step, kernel extension has been studied inspired by the multikernel support vector regression, which has shown its superiority in terms of improving approximation power. As indicated by Zhao and Sun [178], apart from introducing new support vectors, an alternative way to enhance modeling accuracy is to expand the kernel basis functions, like for example adding a constant term and a linear term to the Gaussian kernel. Specifically, kernel extension becomes essential when the local data trend among different subdomains of the system varies greatly.

5.3.1. RECURSIVE KERNEL METHOD USING THE SUPPORT VECTORS

In IRR-LSSVR, a Gaussian kernel was adopted by Zhao et al.[179] with the expression:

$$k(\mathbf{x}_n, \mathbf{x}_m) = exp\left\{-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2}{2\gamma^2}\right\} \tag{5.10}$$

Note that the Gaussian kernels are centered on the selected support vectors as shown in Eq. 5.9, and the outputs of the kernel basis functions $ker(\mathbf{x}_i, \cdot)$ constitute inputs in the feature space rather than outputs. As shown in Eq. 5.9, the weights α are associated with each of the support vectors. If we want to develop a new online fixed-size kernel method for identifying the global model in real-time, we can collect abundant modeling data with enough excitation information at first, and then use IRR-LSSVR to calculate the centers for the parametric kernel method. Having determined the number and the centers of the kernels, we can then use a classical recursive least squares approach to update the weights for each kernel basis functions at each time step. In each iteration, the mapping relationship between the two input spaces remains the same, but the surface shape of the implicit function defined in the high dimensional feature space is always adapted.

In this section, a novel online fixed-size kernel method is developed. This method employs Gaussian kernels as well as the kernel number and positions calculated using IRR-LSSVR. Having chosen the kernel determination technique, we can then realize an online regression framework by applying a classical recursive least squares method. Specifically, the weights α are set as initial weights, which determine the initial shape of

the function output surface in the feature space. Unlike the weights α, the scalar variable b shown in Eq. 5.9 is not updated at each time step. Since the weights of the proposed method vary with time, the method is referred to as weights varying LSSVR (WV-LSSVR) for short.

Before proceeding, we need to have a discussion on the WV-LSSVR. IRR-LSSVR is an offline learning machine, therefore, we need to collect, in advance, enough representing modeling data with full coverage of the system under identification. Besides, the kernel selection method, i.e. IRR-LSSVR, is a nonparametric method, and the scale of the model increases continuously with the increment of the support vector number. Apart from the kernel center selection process, the WV-LSSVR becomes a parametric online kernel method suitable for identifying dynamic global model in real-time. As a result, the WV-LSSVR is a hybrid kernel method, which comprises two processes: kernel center selection process using pre-collected modeling samples and recursive identification of the kernel weights.

5.3.2. EXTENSION OF THE KERNEL BASIS FUNCTION

A widely used kernel function for Gaussian process regression is given by the exponential of a quadratic form, with the addition of constant and linear terms [[28], p.307]:

$$k(\mathbf{x}_n,\mathbf{x}_m) = a_{n0}\cdot exp\left\{-\frac{a_{n1}\|\mathbf{x}_n-\mathbf{x}_m\|^2}{2}\right\} + a_{n2} + a_{n3}\mathbf{x}_n^T\mathbf{x}_m \tag{5.11}$$

where $\mathbf{x}_n$ is the input vector of the current evaluation data and $\mathbf{x}_m$ is one of the support vectors selected offline using IRR-LSSVR. Note that the term involving θ_3 corresponds to a parametric model that is a linear function of the input variables.

By introducing $a_{n1} = \frac{1}{\gamma^2}$, we can get

$$k\left(\mathbf{x}_i,\mathbf{x}_j\right) = \theta_{i1}\cdot exp\left\{-\frac{\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{2\gamma^2}\right\} + \theta_{i2} + \theta_{i3}\mathbf{x}_i^T\mathbf{x}_j \tag{5.12}$$

where $\mathbf{x}_i \in S$, and $S = \{\mathbf{x}_1,...,\mathbf{x}_n\}$ is the selected sub set of support vectors calculated using IRR-LSSVR. Once again, the kernel functions are centered on the selected support vectors, and their bandwidth is chosen the same as those used in IRR-LSSVR. This novel adaptive kernel method will be referred to as Gaussian process kernel based LSSVR (GPK-LSSVR) in the remainder of this chapter. The parameters of the kernel model that needs to be recursively updated has the following expression:
$\mathbf{c} = [\theta_{11},\theta_{12},\theta_{13},\cdots,\theta_{i1},\theta_{i2},\theta_{i3},\cdots,\theta_{M1},\theta_{M2},\theta_{M3}]^\top$, where $i \in [1,\cdots,M]$ and M is the number of support vectors.

GPK-LSSVR includes two stages: kernel determination in an offline manner using pre-collected data and weight updates in real-time using new data samples. The detailed training methodology for GPK-LSSVR is depicted by Algorithm 1. Note that the algorithmic flow chart for WV-LSSVR is similar to Algorithm 1. The only difference lies in step.3. WV-LSSVR keeps using the nominal Gaussian kernel functions shown in Eq. 5.10 instead of extending it into compound kernel functions shown in Eq. 5.12.

Algorithm 1. GPK-LSSVR

step.1 Based on the a priori experience of the model, gather and select offline input-output training data set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ with as large a coverage of the input space as possible.

step.2 Choose the optimal support vector set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{M}$ from $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ and calculate the unknown parameters α_S, b as shown in Eq. 5.8 using IRR-LSSVR from [179].

step.3 Extend the nominal Gaussian kernel shown in Eq. 5.10 into the kernel shown in Eq. 5.12. The kernel centers are determined by the support vector set $\{(\mathbf{x}_i, d_i)\}_{i=1}^{M}$. The initial value of $\theta_1 = [\theta_{11}, \theta_{21}, \cdots, \theta_{M1}]$ is determined by α_S, and the initial values of θ_2 and θ_3 are set to be zero.

step.4 Update the parameters θ_1, θ_2 and θ_3 using recursive least squares method when evaluating new testing data. Execute step.4 recursively for each new data.

5.3.3. COMPUTATIONAL COMPLEXITY

According to [179], IRR-LSSVR has a computational complexity of $\mathcal{O}\left(M \cdot N^2\right)$ in time at each iteration when selecting support vectors, where M is the number of selected support vectors and N is the total number of modeling data. Because WV-LSSVR and GPK-LSSVR both use IRR-LSSVR to determine the kernel centers, see step.2 of Algorithm 1, they enjoy the same computational complexity in time as IRR-LSSVR at each iteration while determining the kernel centers. The second part of the computational complexity of WV-LSSVR and GPK-LSSVR is caused by the recursive updating of the kernel model using newly available data. This part of computational complexity is tabulated in Table 5.1 with m the dimension of the input variables. As shown in Table 5.1, the computational complexity of both methods comprises two parts: kernel basis regression vector computation $\mathcal{O}(m \cdot M)$, and the parameter update using the recursive least squares method $\mathcal{O}\left(M^2\right)$ or $\mathcal{O}\left((3M)^2\right)$. According to Table 5.1, the computational complexity in evaluating a new data will increase quadratically if the number of support vectors increases.

Table 5.1: Computational complexity in each time step

algorithms	time	memory
WV-LSSVR	$\mathcal{O}\left(m \cdot M + M^2\right)$	$\mathcal{O}\left(m \cdot M + M^2\right)$
GPK-LSSVR	$\mathcal{O}\left(m \cdot M + (3M)^2\right)$	$\mathcal{O}\left(m \cdot M + (3M)^2\right)$

5.4. EXPERIMENTS AND RESULTS

To demonstrate the feasibility and efficiency of WV-LSSVR and GPK-LSSVR methods, we will carry out a set of 14 experiments using benchmark data sets and a data set from a real-world object. Among the benchmark data sets, motor-UPDRS, total-UPDRS, winequality-red, winequality-white, concrete, autoMPG, abalone, Boston housing are from the well-known UCI repository [1], and stock, delta elevators, delta ailerons, kine-

[1] Available from URL:http://archive.ics.uci.edu/ml/.

matics, cpu-act, puma8NH are found from the ltorgo collection[2]. Detailed information on these data sets has already been provided on the websites given in the footnotes. In addition, one data set from a real-world system is also used to validate the methods, that is, the data from the diesel engine[16]. Note that all these data sets have also been applied in [179]. All experiments are done on an Intel i5 CPU E31270 (3.40 GHz) processor with 16.00 GB RAM in a Matlab2010b environment.

The root mean squared errors (RMSE) and normalized mean squared errors (NMSE) criteria are used to evaluate the approximation power of the proposed methods, where the RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(\hat{d}_i - d_i\right)^2}{N}} \tag{5.13}$$

The NMSE is defined as:

$$NMSE = \frac{1}{\triangle^2 N} \sum_{i=1}^{N} \left(d_i - \hat{d}_i\right)^2 \tag{5.14}$$

where

$$\triangle^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(d_i - \bar{d}\right)^2$$

with $\bar{d}$ the mean of the measured values, $\hat{d}_i$ the predicted value, and d_i the measured value.

5.4.1. COMPARISON WITH THE K-MEANS CLUSTERING APPROACH

For comparison purpose, a classical k-means clustering (KMC) based RBF kernel method [13, 28] was also implemented. Two new online kernel methods proposed in this chapter, i.e., WV-LSSVR and GPK-LSSVR, and the abovementioned KMC-RBF were compared to each other with special focus on the optimality of the selected kernel centers. The aforementioned benchmark data sets were applied to validate the performance of the kernels chosen using IRR-LSSVR and the KMC method separately. As a first step in the numerical experiments, the training data set was utilized to calculate the number and centers for all the kernel basis functions. As a second step, the same training data set was used again to train the model in a recursive manner. As a third step, the identified model from the previous process was evaluated using the testing benchmark data sets without further updates of the model parameters. Note that all the kernel parameters are selected the same as those used in IRR-LSSVR [179].

To depict the effects of the k-means clustering threshold σ and the kernel number on the approximation accuracy of KMC-RBF, the 'total UPDRS' data set from the UCI repository is selected as validation data set at first. The simulation results are plotted in Figure 5.1 and Figure 5.2. As can be seen from Figure 5.1, when the threshold σ decreases, RMSE also decreases. In addition, once σ is smaller than some value, i.e., 0.01, the influence of its reduction on the approximation accuracy becomes negligible. In Figure 5.1, the RMSE first decreases, then increases, and gets its minimum around 750. Figure 5.2 gives the same trend information on NMSE.

[2] URL:http://www.liaad.up.pt/ ltorgo/Regression/DataSets.html.

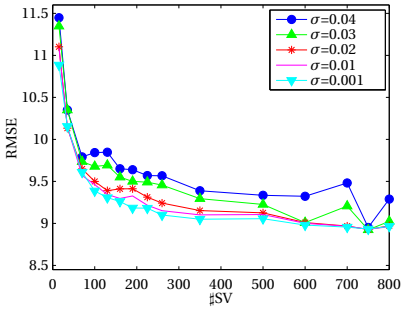


Figure 5.1: RMSE of KMC-RBF, under different σ, total UPDRS.

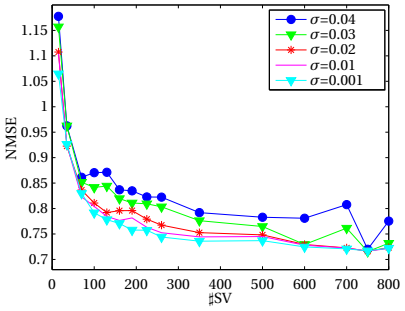


Figure 5.2: NMSE of KMC-RBF, under different σ, total UPDRS.

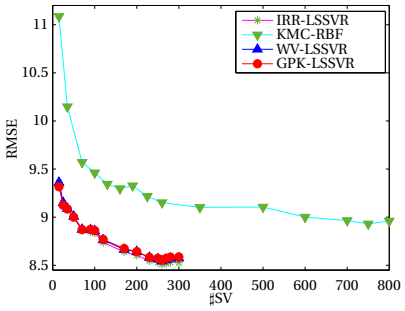


Figure 5.3: RMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$, $\sigma = 0.01$.

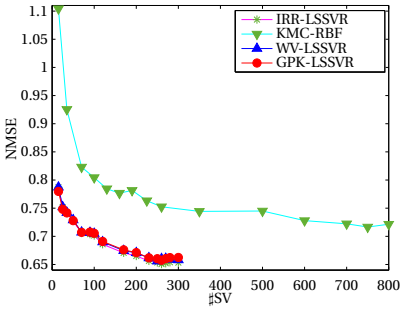


Figure 5.4: NMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$, $\sigma = 0.01$.

Figure 5.3 and Figure 5.4 give the comparison results of KMC-RBF, GPK-LSSVR and WV-LSSVR basing on the 'total UPDRS' data set. Although IRR-LSSVR is an offline method, it is utilized to provide a baseline for the comparison results of the abovementioned methods. The effects of different kernel number and different kernel centers on the approximation power are demonstrated. As can be seen from Figure 5.3, GPK-LSSVR enjoys the same level of approximation accuracy as both WV-LSSVR and IRR-LSSVR, while it has a far more higher approximation accuracy than KMC-RBF. Due to similarity and limitation of space, except for 'total UPDRS', the comparison results for all of the other benchmark data sets are tabulated in Table 5.2. In all of the following numerical experiments, σ is selected as 0.01, and other parameters for IRR-LSSVR are chosen the same as those in [179].

In Table 5.2, seTime denotes the consumed time for the kernel selection process, trTime is the time for identifying the global model recursively, and dimension means the dimensionality of the functional input vector. Additionally, teTime represents the evaluation time using the test samples without model adaptation, while trNum and

teNum stand for the number of training data points and testing data points separately. Last but not the least, ♯SV gives the number of support vectors. The numbers of the support vectors are chosen the same as those used in [179].

According to Table 5.2, both WV-LSSVR and GPK-LSSVR enjoy a lower RMSE when compared to the KMC-RBF method. This indicates that the kernels used in both WV-LSSVR and GPK-LSSVR methods, which are determined using IRR-LSSVR, have higher approximation power than the kernels used in KMC-RBF method, which are calculated using the KMC algorithm. When compared with WV-LSSVR, GPK-LSSVR leads to more or less the same global approximation accuracy.

Furthermore, the seTime of IRR-LSSVR is comparable to, even if it is not always smaller than, that of the KMC.

5.4.2. RECURSIVE IDENTIFICATION RESULTS ON EXTENDED KERNEL BASIS FUNCTIONS

Indicated by [178], multikernel SVMs have demonstrated superiority in enhancing the approximation power, especially when the local data trend in one sub domain differs greatly from that in another sub domain. In view of this, the effects of extending the kernel basis functions are explored in a recursive model identification framework. To illustrate the benefits of the extended kernel basis function, the GPK-LSSVR method is compared with WV-LSSVR, and the results are tabulated in Table 5.3. This experiment consists of two steps. At the first step, IRR-LSSVR was utilized to calculate the kernel centers for both WV-LSSVR and GPK-LSSVR using the benchmark training data sets. Subsequently, both of the preceding methods were employed to learn and evaluate the testing benchmark data sets at the second step. Note that, the validation results shown in this section are different from those shown in section 4.1. That is, updates of the model parameters are involved in the test phase while we were deriving simulation results for this section. Besides, it should be kept in mind that GPK-LSSVR and WV-LSSVR use the same kernel centers as those of IRR-LSSVR, which warrants a fair comparison.

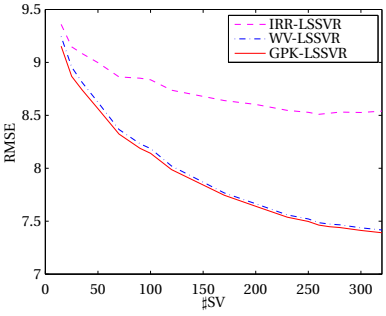


Figure 5.5: RMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$.

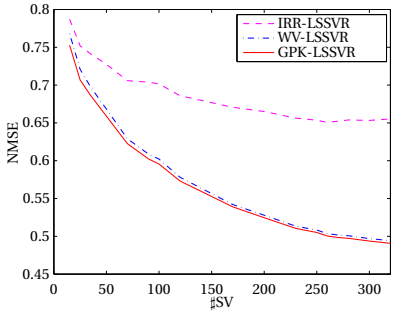


Figure 5.6: NMSE for total UPDRS, $\gamma = 2^5$, $C = 2^{-2}$.

Figure 5.5 and Figure 5.6 give the comparison results of IRR-RBF, GPK-LSSVR and WV-LSSVR using 'total UPDRS' data set. WV-LSSVR and GPK-LSSVR methods lead to

a higher approximation accuracy than IRR-LSSVR, which is mainly because that the test benchmark data are also involved in updating the model in the former methods. In addition, as can be seen from Figure 5.5, GPK-LSSVR enjoys a higher level of approximation accuracy than WV-LSSVR. This improvement should be ascribed to the extension of the kernel basis function, since the extended kernel allows a fast adaptation to higher degree of nonlinearity in local data trends and provides higher degree of freedom on modeling parameters for optimization. Again, due to similarity and limitation of space, except for 'total UPDRS', the comparison results for all of the other benchmark data sets are tabulated in Table 5.3. In all of the following numerical experiments, kernel parameters are chosen the same as those used in IRR-LSSVR from [179].

In Table 5.3, trTime gives the computational time of selecting kernel centers using IRR-LSSVR, while teTime shows the total recursive identification time on the benchmark testing data sets. Besides, the column marked with baseline RMSE shows the RMSE values of the baseline method (i.e.IRR-LSSVR) when evaluating the testing data without updating the model. As can be seen from the RMSE column, the approximation accuracy of both WV-LSSVR and GPK-LSSVR methods is higher than that of the baseline method, this is mainly because the testing data are also used to update the global model in real-time. Furthermore, if we compare GPK-LSSVR with WV-LSSVR, it is found that the former enjoys a higher approximation accuracy than the latter in recursive identification using the aforementioned parametric kernel methods. In addition, GPK-LSSVR always has a higher approximation accuracy than WV-LSSVR although the magnitude of the improvement varies on different data sets. This is due to the fact that the Gaussian process kernel provides higher degrees of freedom than the single Gaussian kernel for the recursive parametric kernel method, which allows the kernel model to track the different data trends among different sub domains. Finally, it should also be noted that the computational time of GPK-LSSVR remains comparable to that of the WV-LSSVR, which is consistent with the analysis result shown in Table 5.1.

5.5. CONCLUSIONS

In the field of SVMs, convincing results have been achieved on techniques of selecting optimal support vectors for LSSVR. Specifically, the support vector selection technique employed in IRR-LSSVR has shown its superiority to other methods. To achieve the same level of modeling accuracy, IRR-LSSVR needs fewer support vectors than other LSSVR methods.

In this chapter, we propose a general recursive parametric kernel method called WV-LSSVR, which is suitable for identifying global models in real-time. In order to calculate the centers for all kernels, the support vector selection technique from the field of SVMs is used. Instead of directly solving a global optimization problem, WV-LSSVR obtains all the kernel centers by using IRR-LSSVR, where support vectors are chosen in a recursive way. Specifically, IRR-LSSVR is employed to analyze a pre-collected modeling data set so as to acquire a suitable number of support vectors, which are later set to be the centers of the kernels in WV-LSSVR.

For the purpose of improving the approximation accuracy of the global model, especially when considering different data trends among different subdomains, an improved version of WV-LSSVR called GPK-LSSVR is developed by extending the kernel basis functions. A kernel basis function recommended for Gaussian process regression is adopted in GPK-LSSVR. The advantage of the Gaussian process kernel is that it helps to simplify the implementation of the proposed recursive kernel method.

To demonstrate the effectiveness of WV-LSSVR and GPK-LSSVR, a set of numerical experiments are carried out using benchmark data sets. At first, classical KMC-RBF is implemented so as to provide a comparison baseline. Compared with KMC-RBF, the kernel centers calculated using IRR-LSSVR lead to a higher approximation accuracy in identifying the global model. Subsequently, GPK-LSSVR method, in which the Gaussian kernels are extended with a linear term and a constant term, is compared with WV-LSSVR in a recursive identification framework. In this numerical experiment, the testing benchmark data are evaluated using WV-LSSVR and GPK-LSSVR while the recursive identification is running. The results validate the hypothesis that a suitable extension of the kernel basis function helps to enhance the approximation power of the recursive parametric kernel method. In the future, techniques for choosing representative modeling data in advance need to be investigated, because all kernel centers are uniquely determined by pre-collected modeling data.

Table 5.2: Comparison results with the K-means clustering method, **without** updates during validation

data sets	algorithms	RMSE	NMSE	seTime[s]	trTime[s]	teTime[s]	trNum	teNum	♯SV
motor UPDRS $C^\star = 2^5$	KMC-RBF	6.9765	7.5682E-01	216.3110	47.1279	1.7748	3000	2875	350
$\gamma^\star = 2^{-2}$, 40 dimension	WV-LSSVR	6.6354	6.8461E-01	46.3323	48.3603	2.1684	3000	2875	350
	GPK-LSSVR	6.6393	6.8542E-01	46.8159	818.0848	5.3196	3000	2875	350
total UPDRS $C^\star = 2^5$	KMC-RBF	9.1513	7.5247E-01	136.8129	23.0257	1.5288	3000	2875	260
$\gamma^\star = 2^{-2}$, 16 dimension	WV-LSSVR	8.5437	6.5587E-01	30.1862	25.0226	1.5600	3000	2875	260
	GPK-LSSVR	8.5546	6.5753E-01	30.2642	522.6190	3.9312	3000	2875	260
winequality-red $C^\star = 2^5$	KMC-RBF	9.4202E-01	13.6340E-01	0.2808	0.2652	0.0312	1000	599	19
$\gamma^\star = 2^{-1}$, 11 dimension	WV-LSSVR	6.2926E-01	6.0837E-01	1.2324	0.1560	0.0312	1000	599	19
	GPK-LSSVR	5.2162E-01	5.9368E-01	1.2324	0.7956	0.3588	1000	599	19
winequality-white $C^\star = 2^6$	KMC-RBF	7.3611E-01	6.9254E-01	11.4349	3.3072	0.5460	3500	1398	70
$\gamma^\star = 2^0$, 11 dimension	WV-LSSVR	7.1246E-01	6.4875E-01	17.7373	3.4944	0.2808	3500	1398	70
	GPK-LSSVR	7.1283E-01	6.4943E-01	17.8465	15.5845	0.6240	3500	1398	70
concrete $C^\star = 2^6$	KMC-RBF	7.2130	19.192E-02	7.8000	1.4352	0.5616	700	330	125
$\gamma^\star = 2^{-1}$, 8 dimension	WV-LSSVR	5.8509E-01	12.628E-02	1.1700	1.6848	0.2652	700	330	125
	GPK-LSSVR	5.9394	13.013E-02	1.1232	13.7437	0.4212	700	330	125
Boston housing $C^\star = 2^5$	KMC-RBF	4.4051	2.0304E-01	1.7628	0.4992	0.0624	400	106	90
$\gamma^\star = 2^{-1}$, 13 dimension	WV-LSSVR	2.6189	7.1766E-02	0.3432	0.6240	0.0468	400	106	90
	GPK-LSSVR	2.6397	7.2912E-02	0.3432	3.8064	0.0936	400	106	90
delta elevators $C^\star = 2^5$	KMC-RBF	1.4578E-03	3.7374E-01	3.0576	3.0108	0.7644	6000	3517	36
$\gamma^\star = 2^0$, 6 dimension	WV-LSSVR	1.4505E-03	3.7001E-01	56.2384	2.6208	0.7800	6000	3517	36
	GPK-LSSVR	1.4508E-03	3.7017E-01	56.0980	9.0481	1.0452	6000	3517	36
puma8NH $C^\star = 2^5$	KMC-RBF	4.0173	5.1544E-01	41.0751	5.9592	0.9984	4500	3693	90
$\gamma^\star = 2^{-1}$, 8 dimension	WV-LSSVR	3.3622	3.6104E-01	31.5122	5.6940	0.9672	4500	3693	90
	GPK-LSSVR	3.3633	3.6128E-01	34.4606	49.4991	2.1216	4500	3693	90
cpu act $C^\star = 2^8$	KMC-RBF	4.3084	5.8782E-02	790.4415	22.1989	1.7628	5000	3192	200
$\gamma^\star = 2^0$, 21 dimension	WV-LSSVR	2.5893	2.1231E-02	55.2244	22.9477	1.7940	5000	3192	200
	GPK-LSSVR	2.5898	2.1238E-02	66.8464	332.0325	3.7752	5000	3192	200
delta ailerons $C^\star = 2^5$	KMC-RBF	1.6508E-04	3.0721E-01	5.1324	4.3056	0.7332	5000	2129	60
$\gamma^\star = 2^{-1}$, 40 dimension	WV-LSSVR	1.6288E-04	2.9906E-01	37.0658	5.1168	0.8424	5000	2129	60
	GPK-LSSVR	1.6284E-04	2.9891E-01	37.5650	16.4269	0.6552	5000	2129	60
Ablone $C^\star = 2^5$	KMC-RBF	2.5246	5.6284E-01	0.0780	0.1404	0.0312	3000	1177	15
$\gamma^\star = 2^1$, 8 dimension	WV-LSSVR	2.1808	4.2000E-01	11.6377	0.5304	0.0312	3000	1177	15
	GPK-LSSVR	2.1829	4.2080E-01	11.9029	1.8876	0.2808	3000	1177	15
stock $C^\star = 2^{10}$	KMC-RBF	12.3160E-01	3.6105E-02	2.3868	1.0608	0.3432	600	350	100
$\gamma^\star = 2^{-1}$,9 dimension	WV-LSSVR	8.1528E-01	1.5822E-02	0.6864	0.7956	0.2496	600	350	100
	GPK-LSSVR	8.1737E-01	1.5904E-02	0.7020	7.1448	0.6864	600	350	100
AutoMPG $C^\star = 2^{10}$	KMC-RBF	3.1642	1.4336E-01	0.0156	0.0156	0.0060	350	42	11
$\gamma^\star = 2^1$, 7 dimension	WV-LSSVR	2.6098	0.9753E-01	0.1716	0.0624	0.0240	350	42	11
	GPK-LSSVR	2.5320	0.9179E-01	0.1716	0.1248	0.0300	350	42	11
kinematics $C^\star = 2^4$	KMC-RBF	9.8647E-02	14.4530E-02	753.8748	179.9004	3.6504	4500	3693	500
$\gamma^\star = 2^{-1}$, 8 dimension	WV-LSSVR	7.9163E-02	9.3077E-02	166.1255	239.1495	3.9312	4500	3693	500
	GPK-LSSVR	7.9123E-02	9.2981E-02	162.7090	3851.6335	9.4693	4500	3693	500
diesel engine $C^\star = 2^{10}$	KMC-RBF	6.7024E-02	1.1151E-03	0.0156	0.0156	0.0156	210	198	15
$\gamma^\star = 1.69$, 3 dimension	WV-LSSVR	2.2658E-02	1.2743E-03	0.0936	0.0624	0.0158	210	198	15
	GPK-LSSVR	2.2389E-02	1.2442E-03	0.1092	0.1248	0.0514	210	198	15

Table 5.3: Comparison results concerning the extended kernel functions, **with** updates during validation

data sets	baseline RMSE	algorithms	RMSE	NMSE	trTime[s]	teTime[s]	trNum	teNum	♯SV
motor UPDRS $C^\star = 2^5$	6.5859	WV-LSSVR	5.7273	5.1004E-01	37.4035	11.3455	3000	2875	350
$\gamma^\star = 2^{-2}$, 40 dimension		GPK-LSSVR	5.6997	5.0515E-01	44.6163	722.7526	3000	2875	350
total UPDRS $C^\star = 2^5$	8.5098	WV-LSSVR	7.4859	5.0351E-01	30.6386	27.0350	3000	2875	260
$\gamma^\star = 2^{-2}$, 16 dimension		GPK-LSSVR	7.4632	5.0046E-01	28.5638	384.4957	3000	2875	260
winequality-red $C^\star = 2^5$	6.2870E-01	WV-LSSVR	5.7554E-01	5.0892E-01	1.2324	0.3120	1000	599	19
$\gamma^\star = 2^{-1}$, 11 dimension		GPK-LSSVR	5.4157E-01	4.5063E-01	1.0920	0.4992	1000	599	19
winequality-white $C^\star = 2^6$	7.1215E-01	WV-LSSVR	6.7444E-01	5.8136E-01	17.5501	1.2480	3500	1398	70
$\gamma^\star = 2^0$, 11 dimension		GPK-LSSVR	6.6549E-01	5.6603E-01	18.1429	7.5036	3500	1398	70
concrete $C^\star = 2^6$	5.8783	WV-LSSVR	3.2515E-01	3.9001E-02	1.1076	0.7176	700	330	125
$\gamma^\star = 2^{-1}$, 8 dimension		GPK-LSSVR	3.0922	3.5272E-02	1.0608	8.0653	700	330	125
Boston housing $C^\star = 2^5$	2.6354	WV-LSSVR	0.7371	0.5685E-02	0.3900	0.2184	400	106	90
$\gamma^\star = 2^{-1}$, 13 dimension		GPK-LSSVR	0.5896	0.3637E-02	0.4680	1.1700	400	106	90
delta elevators $C^\star = 2^5$	1.4503E-03	WV-LSSVR	1.4293E-03	3.5926E-01	38.4889	0.3900	6000	3517	36
$\gamma^\star = 2^0$, 6 dimension		GPK-LSSVR	1.4223E-03	3.5575E-01	53.7423	4.6488	6000	3517	36
puma8NH $C^\star = 2^5$	3.3616	WV-LSSVR	3.1171	3.1032E-01	33.0566	5.7876	4500	3693	90
$\gamma^\star = 2^{-1}$, 8 dimension		GPK-LSSVR	3.0954	3.0602E-01	31.1690	35.02094	4500	3693	90
cpu act $C^\star = 2^8$	2.5816	WV-LSSVR	2.2841	1.6521E-02	64.4440	17.0821	5000	3192	200
$\gamma^\star = 2^0$, 21 dimension		GPK-LSSVR	2.2462	1.5977E-02	56.1604	179.8536	5000	3192	200
delta ailerons $C^\star = 2^5$	1.6278E-04	WV-LSSVR	1.5366E-04	2.6617E-01	27.2879	0.5148	5000	2129	60
$\gamma^\star = 2^{-1}$,40 dimension		GPK-LSSVR	1.5238E-04	2.6175E-01	38.5946	6.6768	5000	2129	60
Ablone $C^\star = 2^5$	2.1804	WV-LSSVR	2.1292	4.0037E-01	11.2945	0.3120	3000	1177	15
$\gamma^\star = 2^1$, 8 dimension		GPK-LSSVR	2.0843	3.8365E-01	11.6689	0.6396	3000	1177	15
stock $C^\star = 2^{10}$	8.1466E-01	WV-LSSVR	5.155E-01	6.3247E-03	0.4277	0.1093	600	350	100
$\gamma^\star = 2^{-1}$,9 dimension		GPK-LSSVR	4.9072E-01	5.7322E-03	0.7332	4.0560	600	350	100
AutoMPG $C^\star = 2^{10}$	2.6099	WV-LSSVR	2.0339	5.9231E-02	0.1519	0.0624	350	42	11
$\gamma^\star = 2^1$, 7 dimension		GPK-LSSVR	1.8101	4.6914E-02	0.1872	0.0624	350	42	11
kinematics $C^\star = 2^4$	7.8735E-02	WV-LSSVR	5.7416E-02	4.8961E-02	150.0730	153.2242	4500	3693	500
$\gamma^\star = 2^{-1}$, 8 dimension		GPK-LSSVR	5.6991E-02	4.8240E-02	152.8338	3414.4875	4500	3693	500
diesel engine $C^\star = 2^{10}$	2.2034E-02	WV-LSSVR	1.9435E-02	0.93758E-03	0.1092	0.0624	210	198	15
$\gamma^\star = 1.69$, 3 dimension		GPK-LSSVR	1.8844E-02	0.8815E-03	0.1092	0.0624	210	198	15

II

Fault Tolerant Flight Control Theory and Practice

In the work presented in Part I, three new model identification methods are developed for the purpose of providing an online global aerodynamic model for a model-based adaptive flight control unit or a flight envelope protection unit, see Figure 2.1. Although the ultimate goal of these studies is to achieve model-based reconfigurable flight control and online flight envelope protection in case of aircraft structural or actuator failures, the global aerodynamic model identification methods developed in this thesis have not been incorporated into real designs of a flight envelope protection approach or a model-based adaptive flight controller. Some relevant work was done in [156] where the substitution based MVSB method developed in Chapter 3 is used to design a model-based adaptive NDI controller for an F-16 aircraft, but this work is out of the scope of this thesis and therefore is not included.

In Part II, this thesis will focus on developing a new and powerful fault-tolerant flight controller. Model-based reconfigurable control methods have advantages and disadvantages. One evident disadvantage is that it needs an accurate online aerodynamic model, which is hard to achieve during a high maneuvering flight condition or when damage occurs to an aircraft. As an alternative to model-based adaptive control methods, the acceleration measurement-based nonlinear incremental control (AMINC) methods are supposed to have the capability of tolerating large model uncertainties thanks to the nature of an incremental and approximate control strategy. The sensor-based backstepping developed in [46] has shown to be theoretically similar to the regular incremental backstepping control approach in section 7.4.2. In Chapters 6 and 7, two double-loop angular controller developed based on a sensor-based backstepping control approach are presented.

In Chapter 8, a state-of-the-art compound structure related to simplex B-splines is studied. This work is mainly aimed at providing more options for selecting model structures when using multivariate simplex B-splines. The objective is to enhance the modeling accuracy using less spline-structure related unknown parameters. Furthermore, this work is partially aimed at estimating a control effectiveness matrix for a regular incremental backstepping or nonlinear dynamic inversion controller. The new model structure called tensor-product simplex (TPS) B-splines is extended from a single-dimension case into a more general multi-dimension case. In addition, TPS B-splines are used to estimate a control effectiveness matrix to meet the need of a regular incremental nonlinear dynamic inversion controller.

6

HYBRID SENSOR BASED BACKSTEPPING CONTROL APPROACH FOR FAULT-TOLERANT CONTROL

Model-based adaptive flight control relies on an accurate onboard aerodynaic model. The accuracy of the identified model can hardly be guaranteed in some flight conditions, and model identification usually results in a high computational load in the onboard computer. An acceleration measurements-based incremental control approach such as a sensor based backstepping (SBB) control method does not require full real-time model information, instead it uses angular acceleration measurements. Incremental control is able to accomodate large model uncertainties due to its approximating adjustment characteristic. An SBB control method is essentially a nonlinear control approach of the incremental type, which usually shows high-gain characteristic. This method is developed based on a singular perturbation theory and initially developed for approximate dynamic inverion control of a nonlinear nonaffine-in-control system. Unlike other incremental control approaches, the SBB method even does not require real-time knowledge of the control effectiveness matrix. In this chapter, a sensor based backstepping control approach is extended to design a flight controller for the fault-tolerant control purpose. It is first applied to design a body angular rate controller for a Boeing 747-200 aircraft. In this step, control surface redundancies are concerned by exploiting the control allocation technique. Then, it is combined with an angular controller, which is designed using a regular nonlinear dynamic inversion control method. Finally, the double-loop angular controller is augmented into a full autopilot. This autopilot is validated by means of flying the Reconfigurable Control for Vehicle Emergency Relief (RECOVER) benchmark model. In addition, the flight performance of the closed-loop system is also evaluated against the engine separation fault and the

rudder runaway scenario.

Recently, an incremental type sensor based backstepping (SBB) control approach, based on singular perturbation theory and Tikhonov's theorem, has been proposed. This Lyapunov function based method uses measurements of control variables and less model knowledge, and it is not susceptible to the model uncertainty caused by fault scenarios. In this chapter, the SBB method has been implemented on a fixed wing aircraft with its focus on handling structural changes caused by damages. A new hybrid autopilot flight controller has been developed for a Boeing 747-200 aircraft after combining nonlinear dynamic inversion (NDI) with SBB control approach. Two benchmarks for fault tolerant flight control (FTFC), named rudder runaway and engine separation, are employed to evaluate the proposed method. The simulation results show that the proposed control approach leads to a zero tracking-error performance in nominal condition and guarantees the stability of the closed-loop system under failures as long as the reference commands are located in the safe flight envelope.

6.1. INTRODUCTION

Research on previous flight accidents [131] and their corresponding fault tolerant flight control (FTFC) strategies suggests that an aircraft, under many post-failure circumstances, can still achieve a certain level of flight performance with the remaining valid control effectors. However, as a consequence of the structural/actuator failures, the control authority or the safe flight envelope of the aircraft is inevitably limited.

Among all fault scenarios, the incidents categorized as 'loss of control in flight' count for as much as 17% of all aircraft accidents [116, 152], and have received most attention. These kinds of failures can be avoided by taking suitable control strategies [131] as suggested by the results of the Flight Mechanics Action Group 16 (FM-AG16), which is a branch of the Group for Aeronautical Research and Technology in Europe (GARTEUR). For example, an FTFC strategy, which involves a fault detection and isolation (FDI) block and a reconfigurable control block, makes it possible to remove the post-failure aircraft from danger [88, 131].

Much research has been done on FTFC in the past few decades. For the purpose of providing a validation platform for modern FDI and FTFC strategies, 6 fault scenarios have been embedded in the Reconfigurable Control for Vehicle Emergency Relief (RECOVER) benchmark model by the FM-AG 16 group including El Al flight 1862 (i.e.engine separation) and rudder runaway[131].

As suggested by Smaili et al. [131], Alwi and Edwards et al. [6] and Lombaerts and Smaili et al. [87], a powerful and advanced control approach is essential to increase the operational performance of the post-failure aircraft. The chosen control algorithms should have at least two of the following merits: it needs to be robust to the sudden structural changes of the aircraft, not relying on an accurate and full aerodynamic model, or it needs to contain a powerful model identification strategy by itself to provide all of the accurate model information for the FDI and reconfigurable control units in real-time.

A number of FDI methods, as well as reconfiguring control approaches, have been proposed in the literature [6, 119, 135, 163, 173, 174]. More recently, the work of

Lombaerts et al. [87], as a part of the GARTEUR FM-AG 16 program, has provided practical validation results of a piloted adaptive nonlinear dynamic inversion (ANDI) controller on the Simulation, Motion, and Navigation (SIMONA) research simulator (SRS). The kernel of this work is a two-step online identification approach aiming at getting the physical model. In this work, rudder runaway case, El Al flight 1862 fault and stabilizer runaway scenarios were studied. The ANDI rate controller guarantees the stability of the post-failure aircraft and enables the pilot to land the aircraft safely. Thereafter, Alwi and Edwards et al.[6] validated another type of reconfigurable control method on the SRS, which was designed using a model reference sliding mode control method together with a constant control allocation matrix. In this work, only El Al flight 1862 scenario was evaluated. The sliding mode control method, which relies on relatively little information of the failure and the extent of the damage to the airframe, has also proven to be able to guarantee the stability of the closed-loop system subject to a certain class of model uncertainties (i.e.structural and actuator changes) caused by the separation of the right wing engines.

Except for utilizing the potential of the remaining control surfaces, researchers have also studied the feasibility of using the differential thrust in emergencies. In the case of rudder or vertical tail failure, the differential thrust control is an effective way to counteract the yawing moments induced by the stuck rudders and thus allow the aircraft to track heading angle commands [48]. A propulsion-controlled aircraft (PCA) system has been developed by the NASA Dryden Research Center, and was first evaluated on a piloted B-720 simulation [50]. In this PCA system, differential thrust was used as an emergency substitute for failed control surfaces [23] such as vertical tail loss with no rudder authority or rudder runaway case [48, 158]. Further research on the PCA system has been carried out by NASA Dryden and Ames Research Centers[158]. Many simulations and actual flight tests of different flight platforms have been performed.

In this chapter, a sensor based backstepping (SBB) approach, which is capable of coping with aerodynamic model changes induced by the failure scenarios, is extended in its application and validated. In 2007, Hovakimyan et al.[59] proposed an advanced controller for non-affine systems, which involves the singular perturbation theory (singular perturbation theory (SPT)), Tikhonov's Theorem and a backstepping strategy. Thereafter, Falkena and van Oort et al. [45] extended its application and named it as SBB control approach. The SBB method was utilized to design a controller for the aircraft moment equations, and was also evaluated on an aerodynamic system with uncertainties and measurement noises. Indicated by [45, 59], as a result of the backstepping control technique, the system stability can be guaranteed by using Lyapunov functions in this SBB approach. In addition, similar to the incremental NDI flight control scheme, the SBB control approach does not need to adapt to uncertain parameters or unknown model structure, which is essential to most model-based conventional backstepping or NDI control approaches. The adaptation requirement is circumvented by using measurements of state derivatives rather than the full knowledge of the model, which is subject to structural or actuator changes.

The objective of this chapter is to present an alternative reconfigurable control approach to the FTFC. This paper uses the SBB control approach proposed in [45], but the focus is shifted to extending its application to designing a generic attitude controller

for a large civil aircraft and handling structural faults for FTFC purposes. In this chapter, the SBB control law is utilized to design a body angular rate controller, while NDI control laws are adopted to design an outer loop angular controller. El Al flight 1862 scenario and rudder runaway fault case, which are challenging benchmark failure scenarios embedded in the RECOVER benchmark model, are utilized to validate the adaptation ability of the inner rate control loop. In rudder runaway case, the differential thrust of engines is introduced to generate necessary yawing moments in order to counteract the aerodynamic yawing moment produced by the stuck rudders. To make the flight simulation results more convincing, a regular flight path controller is also designed using PID control laws.

In Section 6.2, the validation platform is introduced. Subsequently, the basic body angular rate motion equations and the NDI control method are provided in Section 6.3. Thereafter, the single-loop body angular rate controller based on the sensor based backstepping (SBB) approach, as well as a hybrid NDI/SBB attitude hold/change controller, is presented in Section 6.4. In Section 6.5, the simulation experiment results and the corresponding analysis are provided. Finally, concluding remarks are given by Section 6.6.

6.2. VALIDATION PLATFORM

The RECOVER benchmark model of Boeing 747-200 aircraft has been discussed in detail in [131, 132]. This high-fidelity benchmark model was developed for validating the advanced FTFC techniques, and it contains six benchmark fault scenarios:

1. stuck elevators (with/without turbulence)
2. stuck aileron (with/without turbulence)
3. stabilizer runaway (with/without turbulence)
4. rudder runaway (with/without turbulence)
5. loss of vertical tail
6. El Al failure case (dynamic/static method)

Among all of the failure scenarios, El Al flight 1862 scenario and rudder runaway fault case are the top two challenging cases[87], because they greatly cut down the safe flight envelope of the aircraft. Therefore, only these two fault scenarios will be applied to validate the new sensor based control method proposed in this chapter.

6.2.1. RUDDER RUNAWAY AND ENGINE SEPARATION SCENARIOS

The losses and the remaining functional control surfaces in El Al flight 1862 are summarized as follows.

1. Lost surfaces due to the loss of hydraulic systems: outboard trailing-edge flaps, $\delta_{\text{aor}}, \delta_{\text{sp1}}, \delta_{\text{sp4-5}}, \delta_{\text{sp8-9}}, \delta_{\text{sp12}}, \delta_{\text{eil}}, \delta_{\text{eor}}$.

2. Functional but affected surfaces: horizontal stabilizer (half trim rate), δ_{air}, δ_{ail} (both at half rate), and the lower rudder δ_{rl} (lag).

3. Fully functional surfaces: inboard trailing-edge flaps, $\delta_{\text{sp2-3}}$, the left outboard elevator δ_{eol}, and the right inboard elevator δ_{eir}.

In the rudder runaway case, the rudder deflects to the left limit position, inducing a yawing tendency of the aircraft to the left. Since the aerodynamic blow-down is taken into account in the RECOVER simulation model, the rudder deflection limit of this scenario varies with the airspeed. As a result, the maximum rudder deflection is slightly below 15 deg for an airspeed of around 270 kt and close to 25 deg for an airspeed approaching 165 kt.

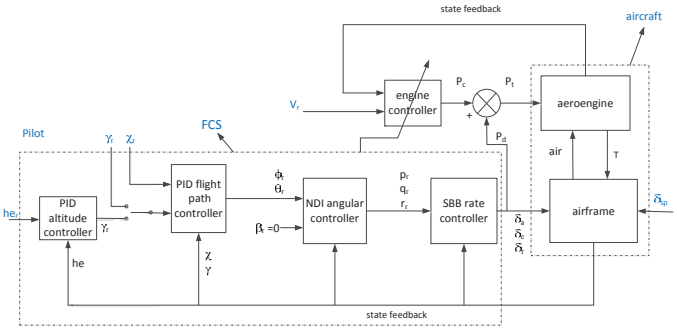
6.2.2. OVERALL AUTOPILOT FLIGHT CONTROL SYSTEM

An autopilot has been designed for the Boeing 747-200 aircraft, which has four control loops as shown in Figure 6.1(a). In the fourth layer of this overall control diagram, an altitude controller is designed using the regular PID control law. In the third loop, a flight path controller has been designed using the regular PID control scheme, where $[\chi, \gamma, V]^\top$ are the controlled variables. $[\phi, \theta, \beta]^\top$ are controlled variables of the angular control loop in the second layer and $[p, q, r]^\top$ are the controlled variables of the first rate control loop. The airspeed is controlled by the collective engine pressure ratio (EPR) P_c. And P_d is the differential EPR used to actively generate a yawing moment. The spoilers are employed to assist the ailerons in order to enhance the control authority of the aircraft in the roll channel.

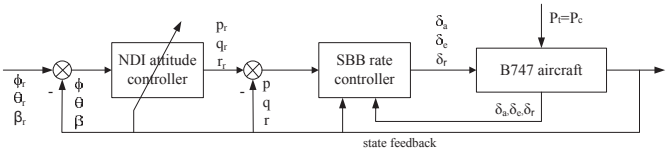
Assuming that the airspeed is able to be governed independently by regulating the engine thrust, the remaining most crucial thing in designing an autopilot flight path controller becomes designing a powerful and reliable angular controller (including rate controller). For the nominal case, i.e., fault-free, and the engine separation failure, a hybrid NDI/SBB angular controller has been designed using regular functional control surfaces without introducing the differential thrust. Compared with the incremental NDI, the advantage of the SBB rate controller is that it does not require the control effectiveness matrix, whose identification values are not adequately trustable during the transient period when sudden structural changes occur to the aircraft. The control structure is given in Figure 6.1(b) with $\mathbf{u} = [\delta_a, \delta_e, \delta_r]^\top$ the control input vector.

To handle the rudder runaway fault, the differential thrust has been introduced to counteract the yawing moment induced by the rudder, and a hybrid NDI/SBB attitude controller has been designed. The structure of this controller is depicted in Figure 6.1(c), where $\mathbf{u} = [\delta_a, \delta_e, P_d]^\top$. Note that, the fault type is assumed to be detectable in the rudder runaway fault scenario.

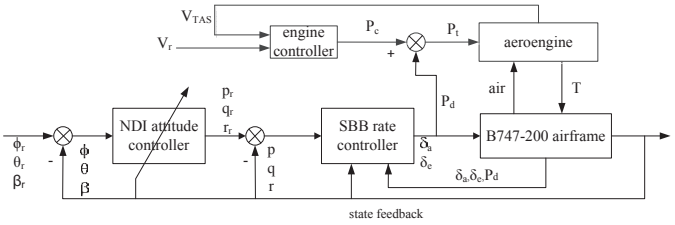
To validate the new flight controller, a simulated flight test benchmark was designed, see Figure 6.2. In this figure, 'F' denotes failure, and Δ indicates an incremental quantity. The simulated flight test is a quite similar flight task to the trajectory tracking assignment carried out by Alwi and Edwards et al. [6], which enables the results in this chapter to be compared to the results presented in [6, 87]. It should be noted that the altitude tracking control task would be switched into flight path angle command tracking mode at the



(a) Autopilot control architecture with four levels.



(b) Attitude controller, engine separation or nominal case.



(c) Attitude controller, rudder runaway.

Figure 6.1: Fault-tolerant controller configuration.

650th second in order to mimic a landing process with fixed gliding slope, i.e., $\gamma = -3$ deg.

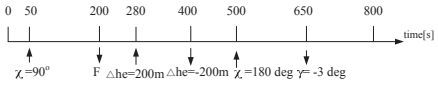


Figure 6.2: Simulated flight test benchmark.

All simulated flight tests in this chapter are started from the same trim point with $V_{TAS} = 133.8$ m/s, altitude=600 m and $\delta_{ih} = -0.65$ deg

.

6.3. ATTITUDE CONTROLLER AND PRELIMINARIES ON RATE CONTROL

6.3.1. ATTITUDE CONTROLLER USING NDI

In order to ensure the angular control performance at a high level, the NDI control law from [87] is utilized to design an attitude controller for Boeing 747-200 aircraft. The reference commands for the inner rate loop are derived from the angular control loop as follows:

$$\begin{bmatrix} p_r \\ q_r \\ r_r \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ \frac{w}{\sqrt{u^2+w^2}} & 0 & \frac{-u}{\sqrt{u^2+w^2}} \end{bmatrix}^{-1} \left(\begin{bmatrix} \nu_\phi \\ \nu_\theta \\ \nu_\beta \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ A_\beta \end{bmatrix} \right) \tag{6.1}$$

with

$$A_\beta = \frac{1}{\sqrt{u^2+w^2}} \left[\frac{-uv}{V^2} \left(A_x - g\sin\theta \right) + \left(1 - \frac{v}{V^2} \right) \left(A_y + g\sin\phi\cos\theta \right) - \frac{vw}{V^2} \left(A_z + g\cos\phi\cos\theta \right) \right]$$

where A_x, A_y, A_z are the acceleration along the body axes without the gravitational effects, and $\begin{bmatrix} \nu_\phi, \nu_\theta, \nu_\beta \end{bmatrix}^\top$ is the virtual angular command vector. The development of A_β is presented in the Appendix Section E. For further details about attitude controller design, the reader can refer to [86].

6.3.2. RATE CONTROL BASIS AND CONTROL ALLOCATION

In order to introduce the control allocation more clearly, it is assumed, in this section, that an NDI rate controller has been designed for the aircraft according to [87]. The control inputs can be solved using the following formulation:

$$\mathbf{M}_{\text{CA}} \cdot \mathbf{u} = \left\{ \frac{\mathbf{I}}{\frac{1}{2}\rho V^2 S} \left(\begin{bmatrix} \nu_p \\ \nu_q \\ \nu_r \end{bmatrix} + \mathbf{I}^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right) - \begin{bmatrix} bC_{l_{\text{states}}} \\ \bar{c}C_{m_{\text{states}}} \\ bC_{n_{\text{states}}} \end{bmatrix} \right\} \tag{6.2}$$

with

$$\mathbf{M}_{CA} = \begin{bmatrix} b & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & b \end{bmatrix} \mathbf{M}_E \tag{6.3}$$

where $\begin{bmatrix} \nu_p \, \nu_q \, \nu_r \end{bmatrix}^\top$ are the virtual rate commands, $\mathbf{M}_{\text{CA}}$ is the control allocation matrix, $\mathbf{M}_E$ is the control effectiveness matrix, $\mathbf{u}$ is the vector consisting of all the control inputs and $C_{l_{\text{states}}}, C_{m_{\text{states}}}, C_{n_{\text{states}}}$ are the non-dimensional moments contributed by all of the current states. In the NDI rate controller, the unknown matrix $\mathbf{M}_E$ and the non-dimensional moments induced by the current states, i.e., $C_{l_{\text{states}}}, C_{m_{\text{states}}}, C_{n_{\text{states}}}$, need to be identified in real-time [87]. One representative aerodynamic model identification method is the two-step identification method [108].

The Boeing 747-200 aircraft has 30 independent control inputs including 25 deflectable control surfaces, four engine pressure ratios (EPRs) and one flight gear (mode) input [131]. To simplify the control allocation logic, some of the aircraft inputs can be combined and the following 19 equivalent control variables can be used instead [5, 6, 158]:

$$\mathbf{u} = \left[\delta_a, \delta_{\text{sp}}, \delta_e, \delta_{\text{ih}}, \delta_r, \mathbf{P}_t, P_d \right]^\top \tag{6.4}$$

with

$$\begin{aligned}
\delta_a &= [\delta_{\text{air}}, \delta_{\text{ail}}, \delta_{\text{aor}}, \delta_{\text{aol}}] \\
\delta_{\text{sp}} &= [(\delta_{\text{sp1}} + \delta_{\text{sp4}}), (\delta_{\text{sp2}} + \delta_{\text{sp3}}), (\delta_{\text{sp10}} + \delta_{\text{sp11}}), (\delta_{\text{sp9}} + \delta_{\text{sp12}})] \\
\delta_e &= [\delta_{\text{eir}}, \delta_{\text{eil}}, \delta_{\text{eor}}, \delta_{\text{eol}}] \\
\delta_r &= [\delta_{\text{ru}}, \delta_{\text{rl}}] \\
\mathbf{P}_t &= [P_{t_1}, P_{t_2}, P_{t_3}, P_{t_4}] \\
P_d &= \frac{1}{4}[(P_{t_1} - P_{t_4}) + (P_{t_2} - P_{t_3})]
\end{aligned} \tag{6.5}$$

with P_c the collective engine pressure ratio (EPR), P_d the differential EPR and $\mathbf{P}_t$ the vector consisting of four total EPRs. They are defined as follows:

$$\begin{aligned}
P_c &= mean(\mathbf{P}_t) \\
P_{t_1} &= P_{t_2} = P_c + P_d \\
P_{t_3} &= P_{t_4} = P_c - P_d
\end{aligned} \tag{6.6}$$

Supposing that the matrix M_E in Eq. 6.2 has been identified and is currently available, an optimizer can be designed to solve the control allocation problem described by Eq. 6.2. However, the overall control effectiveness matrix and thus the control allocation operation may become unreliable during the transient period when structural model changes happen suddenly. In order to enhance the reliability of the control allocation operation in implementing the new control method, the M_E matrix used in this chapter is simplified in further:

$$M_E = \begin{bmatrix} \tilde{C}_{l\delta_a} & 0 & \tilde{C}_{l\delta_r} \\ 0 & \tilde{C}_{m\delta_e} & 0 \\ \tilde{C}_{n\delta_a} & 0 & \tilde{C}_{n\delta_r} \end{bmatrix} \tag{6.7}$$

with

$$\begin{aligned}
\tilde{C}_{l\delta_a} &= -C_{l\delta_{\text{air}}} + C_{l\delta_{\text{ail}}} - C_{l\delta_{\text{aor}}} + C_{l\delta_{\text{aol}}} - \\
&\quad C_{l\delta_{\text{sp1}}} - \cdots - C_{l\delta_{\text{sp4}}} + C_{l\delta_{\text{sp9}}} + \cdots + C_{l\delta_{\text{sp12}}} \\
\tilde{C}_{n\delta_a} &= -C_{n\delta_{air}} + C_{n\delta_{ail}} - C_{n\delta_{aor}} + C_{n\delta_{aol}} - \\
&\quad C_{n\delta_{\text{sp1}}} - \cdots - C_{n\delta_{\text{sp4}}} + C_{n\delta_{\text{sp9}}} + \cdots + C_{n\delta_{\text{sp12}}} \\
\tilde{C}_{m\delta_e} &= C_{m\delta_{\text{eir}}} + C_{m\delta_{\text{eil}}} + C_{m\delta_{\text{eor}}} + C_{m\delta_{\text{eol}}} \\
\tilde{C}_{l\delta_r} &= C_{l\delta_{\text{ru}}} + C_{l\delta_{\text{rl}}} \\
\tilde{C}_{n\delta_r} &= C_{n\delta_{\text{ru}}} + C_{n\delta_{\text{rl}}}
\end{aligned} \tag{6.8}$$

Note that, Eq. 6.8 indicates that the control surfaces belonging to the same category would get equally distributed deflection commands in the control allocation process.

6.4. THE SBB RATE CONTROLLER

In 2002, a singular perturbation theory based nonlinear control method was presented by Khalil et al. [67]. More details on approximate dynamic inversion are included in Appendix H. Then this control law was developed further by Hovakimyan et al. [59] to control a non-affine nonlinear system. In 2011, Falkena et al. [45] combined the singular

perturbation theory with the backstepping technique, and developed an incremental type nonlinear backstepping control approach called the sensor based backstepping (SBB) approach. This Lyapunov function based control method can both guarantee the stability of the closed-loop system and avoid the requirement of full aerodynamic model information [44, 45].

In approximate dynamic inversion, the controller dynamics introduced by an approximation process, see Appendix H, can be viewed as a subsystem cascaded to the body angular rate dynamic system. This allows us to use the singular perturbation theory based SBB control approach to design a body angular rate controller for the Boeing 747-200 aircraft model. The structure of the rate controller is shown in the first level of Figure 6.1(a).

6.4.1. SENSOR BASED BACKSTEPPING RATE CONTROL

The following expression holds for the rotational motion of equations for body angular rates:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = -\left\{ \mathbf{I}^{-1} \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right) - \frac{1}{2} \rho V^2 S \cdot \mathbf{I}^{-1} \begin{bmatrix} bC_{l_{\text{states}}} \\ \bar{c}C_{m_{\text{states}}} \\ bC_{n_{\text{states}}} \end{bmatrix} \right\} + \frac{1}{2} \rho V^2 S \cdot \mathbf{I}^{-1} \mathbf{M}_{\text{CA}} \cdot \mathbf{u} \tag{6.9}$$

Rewrite Eq. 6.9, a simplified formulation of the aircraft motion equations is derived:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{g} \cdot \mathbf{u} \tag{6.10}$$

with

$$\mathbf{x} = \begin{bmatrix} p & q & r \end{bmatrix}^\top \tag{6.11a}$$

$$\mathbf{y}_r = \begin{bmatrix} p_r & q_r & r_r \end{bmatrix}^\top \tag{6.11b}$$

$$\mathbf{e} = \mathbf{x} - \mathbf{y}_r \tag{6.11c}$$

$$\mathbf{g} = \frac{1}{2} \rho V^2 S \cdot \mathbf{I}^{-1} M_{\text{CA}} \tag{6.11d}$$

$$f(\mathbf{x}) = -\mathbf{I}^{-1} \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right) + \frac{1}{2} \rho V^2 S \cdot \mathbf{I}^{-1} \begin{bmatrix} bC_{l_{\text{states}}} \\ \bar{c}C_{m_{\text{states}}} \\ bC_{n_{\text{states}}} \end{bmatrix} \tag{6.11e}$$

In order to design a single-loop body rate backstepping controller, the control Lyapunov function V is chosen as follows:

$$\begin{aligned} V(\mathbf{e}) &= \frac{1}{2} \mathbf{e}^\top \mathbf{e} + \frac{1}{2} \mathbf{k} \lambda^\top \lambda \\ \dot{V}(\mathbf{e}) &= \mathbf{e}^\top \dot{\mathbf{e}} + \mathbf{k} \lambda^\top \mathbf{e} \end{aligned} \tag{6.12}$$

with $\mathbf{e} = \mathbf{e}(t)$, $\mathbf{k}$ a diagonal matrix of controller gains, and $\lambda = \int_0^t \mathbf{e} dt$ an integral term introduced to remove the tracking errors caused by the internal dynamics. Note that $\dot{\lambda} = \mathbf{e}$ holds.

Using Eq. 6.11c, the following expression can be derived for the desired state of the control system:

$$\mathbf{e} = \mathbf{x}_{des} - \mathbf{y}_r \tag{6.13}$$

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_{des} - \dot{\mathbf{y}}_r \tag{6.14}$$

Substituting Eq. 6.14 into Eq. 6.12 results in:

$$\dot{V}(\mathbf{e}) = \mathbf{e}^\top \dot{\mathbf{e}} + \mathbf{k}\lambda^\top \mathbf{e} = \mathbf{e}^\top \left(\dot{\mathbf{x}}_{des} - \dot{\mathbf{y}}_r + \mathbf{k}\lambda\right) \tag{6.15}$$

To stabilize this system, $\dot{\mathbf{x}}_{des}$ can be selected as:

$$\begin{aligned} \dot{\mathbf{x}}_{des} &= -\mathbf{ce} + \dot{\mathbf{y}}_r - \mathbf{k}\lambda \\ &= -\mathbf{c}\left(\mathbf{x} - \mathbf{y}_r\right) + \dot{\mathbf{y}}_r - \mathbf{k}\lambda \end{aligned} \tag{6.16}$$

with $\mathbf{c}$ a positive diagonal matrix to stabilize the system. This yields the desired system:

$$\dot{\mathbf{e}} = -\mathbf{c}\left(\mathbf{x} - \mathbf{y}_r\right) \tag{6.17}$$

The following notation is defined for later usage:

$$\mathbf{u}_{\text{red}} = \mathbf{M}_{\text{CA}} \cdot \mathbf{u} \tag{6.18}$$

with $\mathbf{u}_{\text{red}}$ a three dimensional vector denoting the equivalent inputs. After substituting Eq. 6.18 into Eq. 6.11d, the SBB controller for Eq. 6.10 can be derived according to [45, 59, 67]:

$$\begin{aligned} \epsilon \dot{\mathbf{u}}_{\text{red}} &= -sgn\left(\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}_{\text{red}}}\right)\left[\dot{\mathbf{x}} - \dot{\mathbf{x}}_{des}\right] \\ &= -sgn\left(\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}_{\text{red}}}\right)\left[\dot{\mathbf{x}} + \mathbf{c}\left(\mathbf{x} - \mathbf{y}_r\right) - \dot{\mathbf{y}}_r + \mathbf{k}\lambda\right] \end{aligned} \tag{6.19}$$

where ϵ is a tuning parameter with a small positive value, i.e., $0 < \epsilon \ll 1$. From Eq. 6.9, the following formulation can be obtained:

$$-sgn\left(\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}_{\text{red}}}\right) = -sgn\left(\frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\right) \tag{6.20}$$

Substituting Eq. 6.20 into Eq. 6.19, the control inputs are computed as follows:

$$\dot{\mathbf{u}}_{\text{red}} = -\frac{1}{\epsilon} sgn\left(\frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\right)\left[\dot{\mathbf{x}} + \mathbf{c}\left(\mathbf{x} - \mathbf{y}_r\right) - \dot{\mathbf{y}}_r + \mathbf{k}\lambda\right] \tag{6.21}$$

Using integration, the equivalent control input $\mathbf{u}_{red}$ can be calculated as follows:

$$\mathbf{u}_{\text{red}_k} = \mathbf{u}_{\text{red}_{k-1}} + \int_{(k-1)T}^{kT} \dot{\mathbf{u}}_{\text{red}} \cdot dt \tag{6.22}$$

According to Eq. 6.18, the control input $\mathbf{u}$ can be solved using a control allocation algorithm if $\mathbf{M}_{\text{CA}}$ is available.

Note that, the objective of this chapter is to present a flight controller which does not require an online aerodynamic model. However, Eq. 6.18 is still dependent on the partial aerodynamic model, i.e., a control effectiveness matrix $\mathbf{M}_{\text{CA}}$) identified in real-time. In order to remove this online model dependency, a fixed $\mathbf{M}_{\text{CA}}$ matrix is usually used (see. [6]). The drawback of doing so is that the optimality of the control allocation can not

be guaranteed since it is directly determined by the accurate knowledge of $\mathbf{M}_{\text{CA}}$. In this chapter, a fixed $\mathbf{M}_{\text{CA}}$ is assumed to be available at a trim point.

To simplify the implementation of the controller, the requirement of the control allocation is removed from the design procedures. Remember that the control surfaces have been categorized into 3 groups (see. Eq. 6.8). In each group, a control surface is chosen as the representative control input. Consequently, a representative control input vector $\mathbf{u}_{\text{rep}} = [\delta_{1_{\text{rep}}}, \delta_{2_{\text{rep}}}, \delta_{3_{\text{rep}}}]$ can be derived with $\delta_{1_{\text{rep}}}, \delta_{2_{\text{rep}}}, \delta_{3_{\text{rep}}}$ a representative control surface deflection selected from each category respectively. For example, the representatives can be selected as follows: $\delta_{1_{\text{rep}}} = \delta_{\text{ail}}$, $\delta_{2_{\text{rep}}} = \delta_{\text{eil}}$ and $\delta_{3_{\text{rep}}} = \delta_{\text{ru}}$ (or $\delta_{3_{\text{rep}}} = P_d$). The control allocation matrix $\mathbf{M}_{\text{CA}_{\text{rep}}}$ can be calculated from the aforementioned fixed matrix $\mathbf{M}_{\text{CA}}$ using Eq. 6.7. This means there exists the following assumption at this place:

$$\mathbf{M}_{\text{CA}_{\text{rep}}} \cdot \mathbf{u}_{\text{rep}} \approx \mathbf{M}_{\text{CA}} \cdot \mathbf{u} \tag{6.23}$$

After substituting Eq. 6.23 into Eq. 6.11d, the SBB controller can be redesigned for the system Eq. 6.10. Consequently, the term $\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}_{\text{red}}}$ in Eq. 6.19 and Eq. 6.20 should be replaced by $\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}_{\text{rep}}}$, and Eq. 6.20 becomes:

$$-sgn\left(\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}_{\text{rep}}}\right) = -sgn\left(\frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\mathbf{M}_{\text{CA}_{\text{rep}}}\right) \tag{6.24}$$

While, Eq. 6.21 becomes:

$$\dot{\mathbf{u}}_{\text{rep}} = -\frac{1}{\epsilon} sgn\left(\frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\mathbf{M}_{\text{CA}_{\text{rep}}}\right)\left[\dot{\mathbf{x}} + \mathbf{c}\left(\mathbf{x} - \mathbf{y}_r\right) - \dot{\mathbf{y}}_r + \mathbf{k}\lambda\right] \tag{6.25}$$

Subsequently, $\mathbf{u}_{\text{rep}}$ can be calculated using Eq. 6.22. In calculating $sgn\left(\frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\mathbf{M}_{\text{CA}_{\text{rep}}}\right)$, the same method as that used in [59] is adopted. That is, only the sign of the diagonal elements of the matrix (in the bracket) are used in designing the body angular rate controller.

The configuration of the SBB rate controller is summarized in Figure 6.3. 'TA' is the

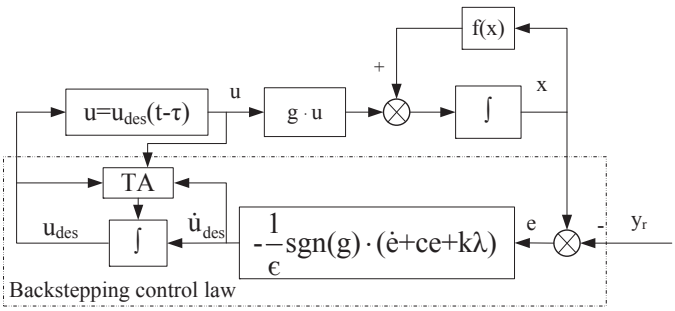


Figure 6.3: flow chart of the SBB inner-loop controller.

acronym of tuning algorithm (TA), and τ represents the time delay between the achieved

control input $\mathbf{u}$ and the controller output $\mathbf{u}_{\text{des}}$, which is caused by the dynamics, saturation or failures of the actuators. It also should be noted that the SBB method contains a tuning parameter called time-scale constant parameter denoted by ϵ, which helps to simplify the tuning process of other control gains.

6.4.2. COMMAND FILTER AND INTEGRATION SATURATION

A command filter is designed to regulate the given reference commands in order to enhance handling qualities of the closed-loop aircraft system. By regulating the reference command into an achievable command, the command filter can play a crucial role in preventing the aircraft from leaving the safe flight envelope.

The windup effect associated with the integrator needs to be removed. The saturation effect may become even severe when some structural failures happen to the control surfaces or there exists a big time-delay on the control effector, e.g., the propulsion system. In this chapter, a tuning algorithm (TA) block was designed to prevent the closed-loop system from integral windup. It uses the discrepancy information between the achieved control inputs $\mathbf{u}$ and the integrator outputs $\mathbf{u}_{\text{des}}$. As shown in Figure 6.3, the TA block will compare $\mathbf{u}_{\text{des}}$ with $\mathbf{u}$ and the saturated position limits. If the actuators are saturated or the changing rate of $\mathbf{u}$ is far more slower than $\mathbf{u}_{\text{des}}$, the integration operation, which intends to increase the difference (determined by the sign of $\dot{\mathbf{u}}_{\text{des}}$), will be skipped in the current time instant.

6.4.3. SLIDING MODE DIFFERENTIATION

In [78], Levant et al. designed a sliding mode diffentiator. The sliding mode method guarantees a precise differentiation under measurement noises. In our work, we firstly get the body angular rates from an extended Kalman filter. Then we use a sliding mode differentiator to calculate the body angular accelerations $\dot{p}$, $\dot{q}$, $\dot{r}$ for Eq. 6.19.

6.5. RESULTS AND ANALYSIS

Up to now, a hybrid NDI/SBB angular control approach has been developed in Section 6.3 and Section 6.4. In order to allow the aircraft to track the flight path commands (χ_r, γ_r and V_{TASr}), the airspeed controller and the flight path angle controller are designed to complete the autopilot designing. Cooperative control of flight path angle and airspeed can be achieved using the total energy control principles (TECS) [168] or the model-based dynamic inversion method [84], which takes into account couplings of flight dynamics. Both of these two methods have the potential to enhance the airspeed and the flight path control performance. However, the focus of this chapter is limited to validating the proposed hybrid NDI/SBB angular controller, which does not require any online model information. In this chapter, a flight path controller is designed using the regular PID control laws, where γ, χ are regulated independently (see. Figure 6.1(a)). In addition, an independent airspeed controller is designed using PID, where P_c are chosen as control inputs. It should be noted that the flight path control-loop does not require the adaptation for changes in aerodynamic forces, e.g., lift, drag and side-force.

The overall flight path controller will be validated using the aerodynamic model

Table 6.1: Control input units and maximum values

Control	Unit	Min	Max	Full hydraulic rate	Half hydraulic rate
δ_{ai}	deg	-20	20	+40/-45 deg/s	+27/-35 deg/s
δ_{ao}	deg	-25	15	+45/-55 deg/s	+22/-45 deg/s
δ_e	deg	-23	17	±37 deg/s	+30/-26 deg/s
δ_r	deg	-25	25	±50 deg/s	±40 deg/s
δ_{ih}	deg	-12	3	±0.2-±0.5 deg/s	±0.1-±0.25deg/s
$\delta_{sp1-4}, \delta_{sp9-12}$	deg	0	45	+75 deg/s	0
δ_{sp5} δ_{sp8}	deg	0	20	+75 deg/s	0
δ_{sp6} δ_{sp7}	deg	0	20	+25 deg/s	0
P_{t1-4}	-	0	1.62	± 0.2 s^{-1}	± 0.2 s^{-1}

Table 6.2: Command filter parameters and command ranges

Commands	Unit	Min	Max	ω_n	ζ
p	rad	-0.2	0.2	6 rad/s	1
q	rad	-0.2	0.2	3 rad/s	1
r	rad	-0.1	0.1	3 rad/s	1
ϕ	deg	-20	20	2.5 rad/s	1
θ	deg	-12	12	2.5 rad/s	1
β	deg	-20	20	2.5 rad/s	1

of Boeing 747-200 aircraft. It will be firstly evaluated for the nominal case and then evaluated using rudder runaway and right engine separation failures introduced in Section 6.2. Eq. 6.8 and Eq. 6.6 are implemented to realize the control allocation, i.e., equally distributed.

6.5.1. COMMAND FILTER SETUP AND ACTUATOR WORKING RANGE

The actuators of the control surfaces are modeled with saturation limits and deflection rate limits (see Table 6.1). In this chapter, a command filter developed in [47] would be utilized. This filter has an adjustable natural frequency ω_n and damping ratio ζ. The scheduling limits on the body angular rate commands and the attitude angular commands are listed in Table 6.2.

6.5.2. OUTER LOOP CONTROLLER PARAMETERS

As mentioned in Sec.6.2, a PID controller has been designed to control V_{TAS}, χ, γ and he. The PID parameters of these outer loop controller are listed in Table 6.3. It should be noted that a sine function with the following formulation has been employed in designing the altitude controller:

Table 6.3: PI parameters of the outer loop PID controllers

Channel	$\chi \leftarrow \phi$	$\gamma \leftarrow \theta$	he $\leftarrow \gamma$	$V_{\text{TAS}} \leftarrow P_c$
Proportional gain	2	0.7	0.139	0.02
Integral gain	0.1	0.3	0	0.0025

Table 6.4: Hybrid NDI/SBB attitude controller parameters, $\epsilon = 0.15$, nominal/engine separation

Channel	proportion	integration
Angular control	[1, 0.5, 1]	[0, 0, 0]
Body rate control	[0.1, 0.2, 0.1]	[0, 0, 0]

$$\gamma_r = \arcsin\left(\frac{\dot{h}e_{\text{cmd}}}{V_{\text{TAS}}}\right) \tag{6.26}$$

where $\dot{h}e_{\text{cmd}}$ denotes the virtual (i.e.commanded) derivative of the altitude, i.e., the output of the altitude PID controller.

6.5.3. VALIDATION RESULTS OF THE NOMINAL AIRCRAFT

The proposed hybrid NDI/SBB attitude controller, see Figure 6.1(b), is tested by flying the Boeing 747-200 aircraft in the nominal state, i.e., fault-free case. The numerical simulation results are plotted in Figs. 6.4-6.15. The idea of a fault-free test of the controller is to show the capability of the proposed controller. The chosen values of the controller parameters are listed in Table 6.4.

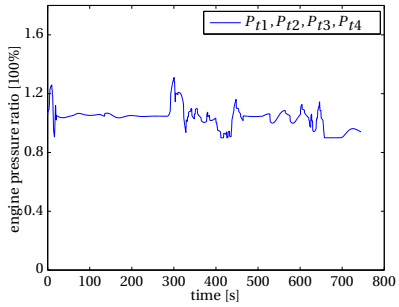


Figure 6.4: Engine Pressure Ratios, nominal.

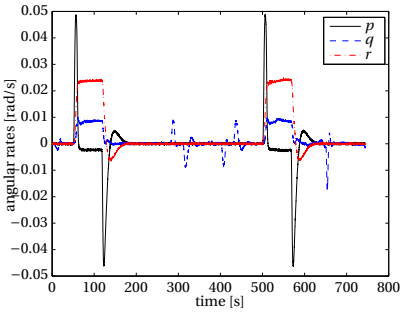


Figure 6.5: Angular rates, nominal.

Figs. 6.4-6.15 give the validation results of the controller under the nominal flight case. The designed control command sequences, see Figure 6.2, were fed to the autopilot. Figure 6.4 shows the changing history of the EPRs, and the EPRs are regulated to keep the true airspeed between 142 m/s and 122 m/s throughout the simulation, see Figure 6.6.

The body angular rate changes are illustrated in Figure 6.5, and the tracking performance of the angular commands is depicted in Figs. 6.7-6.9. As can be seen from Figs. 6.7-6.9, the inner component, i.e., attitude controller, of the autopilot controller enables the aircraft to closely track the attitude commands.

Figs. 6.10-6.12 show the commanded and actual deflections of aileron, elevator and rudder respectively. Specifically, Figure 6.10 shows the commanded and actual aileron deflections. As can be seen from them, the actual control surface deflections

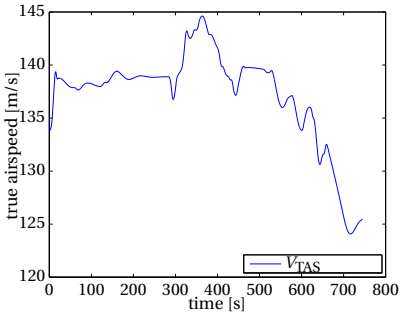


Figure 6.6: True airspeed, nominal.

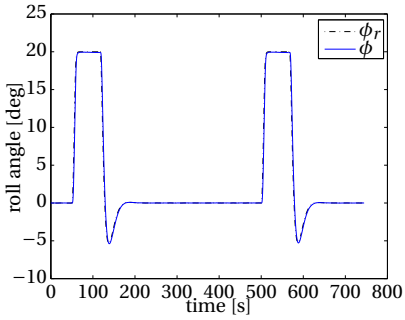


Figure 6.7: Roll angle, nominal.

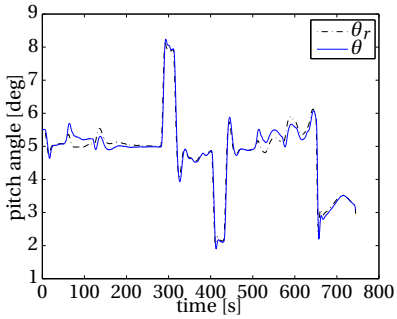


Figure 6.8: Pitch angle, nominal.

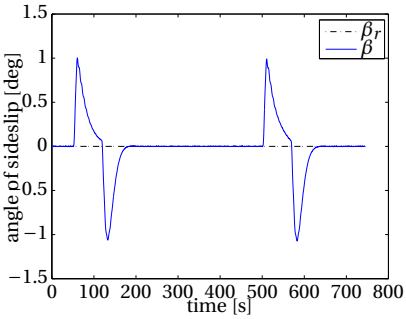


Figure 6.9: Angle of sideslip, nominal.

highly match the commanded deflections, which also means there exists no integration saturation.

Finally, the tracking performance in the flight path control level are illustrated in Figs. 6.13-6.15. The tracking commands of he, γ and χ are well followed. In all simulation experiments of this chapter, the aircraft is in the altitude control mode before the 650th second. And the altitude control loop is switched into the flight path angle control mode at the 650th second. The three-dimensional trajectory is shown in Figure 6.16.

6.5.4. VALIDATION RESULTS USING FAULT SCENARIOS

In the first simulation experiment, the hybrid attitude controller shown in Figure 6.1(b) is evaluated. The controller parameters have already been listed in Table 6.4.

The simulated flight test results of the right engine separation scenario are plotted in Figs. 6.17-6.28. The engine separation failure is triggered at the 200th second, see Figure 6.17.

Figure 6.17 shows the EPRs of the remaining working engines (engine ♯1 and engine ♯2). The EPRs are adjusted to keep V_{TAS} around 140 m/s, see Figure 6.19, and try to slow down the airspeed before landing. It should be noted that the true airspeed shows a decrease fight right after 280th second due to the inadequate supply of power during the climbing. The changing histories of the actual values of p, q, and r are illustrated in

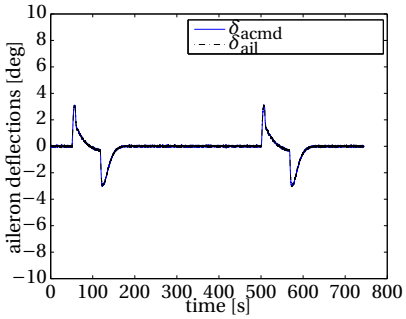


Figure 6.10: Commanded and actual aileron deflections, nominal.

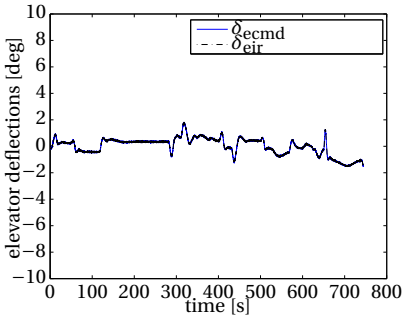


Figure 6.11: Commanded and actual elevator deflections, nominal.

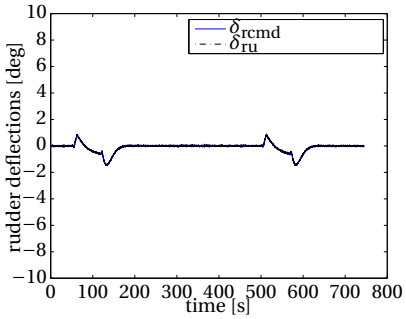


Figure 6.12: Commanded and actual rudder deflections, nominal.

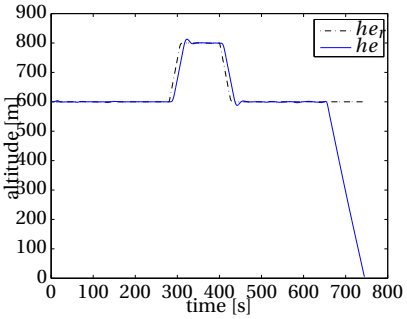


Figure 6.13: Altitude, nominal.

Figure 6.18, and they show stability character even under the engine separation failure. During the coordinate turning, see Fig. 6.20, from 50s to 120s, p and r are regulated cooperatively. While, they are all kept around zero during level straight flight.

The attitude command tracking performance of the proposed controller is depicted in Figs. 6.20-6.22. The figures clearly show that ϕ and θ have a zero tracking error , β will keep smaller than 1.8 deg. The tracking error of β decreases slowly, this is because the remaining control authority of the rudders are quite limited. This is due to the fact that a relatively large part of the working range of the rudders has been occupied by the requirement of compensating the yawing moment produced by the right wing engine separation as can be seen in Figure 6.25.

The changing history of the commanded and actual (under the physical limitations) control surface deflections are depicted by Figs. 6.23-6.25. The control surface deflections of aileron, elevator and rudder are plotted separately. As can be seen from Figs. 6.23-6.24, δ_{eil} δ_{eor} and δ_{aor} are not active/responding under this failure scenario, which is consistent with the description of the engine separation scenario given by Section 6.2. In Figure 6.25, there exists large difference between the commanded and the actual rudder deflection after the 200$^{\text{th}}$ second, which indicates that the rudder often, e.g.,

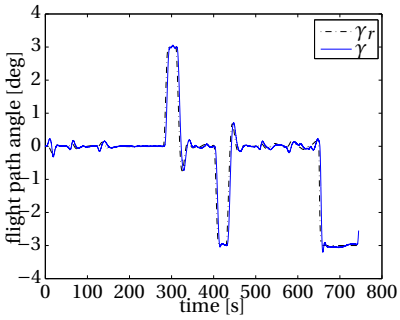


Figure 6.14: Flight path angle, nominal.

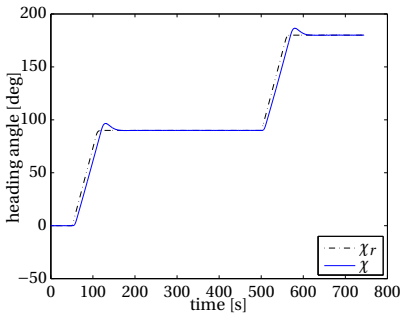


Figure 6.15: Heading angle, nominal.

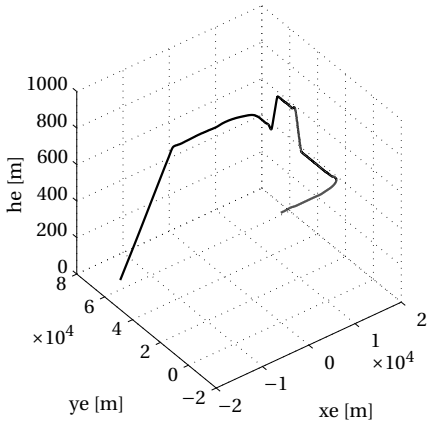


Figure 6.16: Three-dimensional trajectory, nominal.

200^{th} 400^{th} seconds, needs to work in a saturated state.

Figs. 6.26-6.28 provide the records of the command tracking performance in the flight path level. The heading angle command as well as the altitude command has been well tracked. It should be mentioned that the aircraft is under the altitude control mode before the 650^{th} second. Therefore, the flight path angle controller is acting as an inner-loop controller, which is thus not necessary to remove the transient tracking errors. As indicated by [6, 87], the climbing capability would be greatly reduced in the engine separation scenario. That is, a deep climbing becomes not achievable without airspeed loss even when the engine thrust is saturated. In our simulation, the aircraft climbs from $H = 600$m to $H = 800$m in about 50 seconds. The airspeed loss is about 10m/s, which is the price paid for the climbing. Finally, the three-dimensional trajectory is shown in Figure 6.29, where the curve for the nominal case is borrowed from Figure 6.16.

The hybrid NDI/SBB attitude controller is validated using the rudder runaway scenario. As shown in Figure 6.1(c), differential thrust is utilized to compensate the

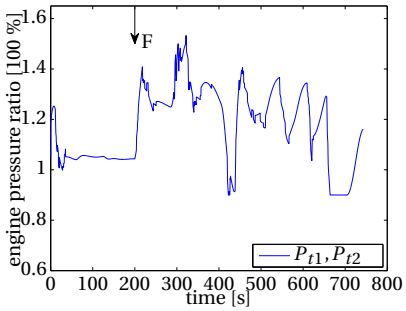


Figure 6.17: Engine Pressure Ratios, engine separation.

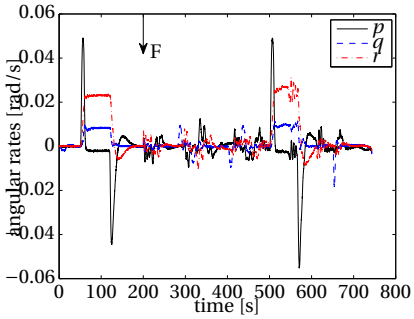


Figure 6.18: Angular rates, engine separation.

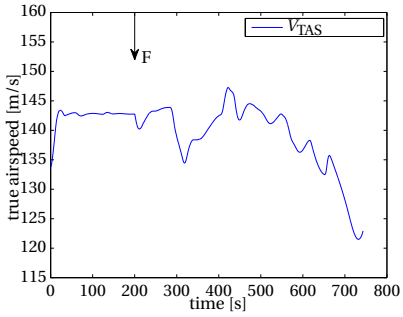


Figure 6.19: True airspeed, engine separation.

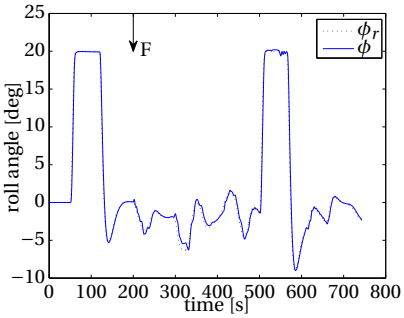


Figure 6.20: Roll angle, engine separation.

yawing moment induced by the failure. The controller parameters are tabulated in Table 6.5. The rudder runaway test of the hybrid controller is also performed for showing the adaptation ability of the proposed controller when the aircraft model changes suddenly.

The validation results of the hybrid NDI/SBB attitude controller in the rudder runaway case are plotted in Figs.6.30-6.43. The changes of the total EPRs ($\mathbf{P}_t$) are shown in Figure 6.30, and all of them reach saturation limits just after the rudder runaway failure occurs. The true airspeed is controlled by the collective thrust P_c, and its changing history is plotted in Figure 6.32. V_{TAS} ranges from 135 m/s to 160 m/s. This is made possible by limiting the upper bound of the total thrust P_t when the rudder is stuck to the left limit. The differential thrust P_d is regulated to actively generate yawing moment once needed by the flight control mission.

Table 6.5: Hybrid NDI/SBB attitude controller parameters, $\epsilon = 0.35$, rudder runaway

Chanel	proportion	integration
Angular control	[1, 1, 1]	[0.12, 0.12, 0.02]
Body rate control	[2, 1, 0.1]	[0, 0, 0]

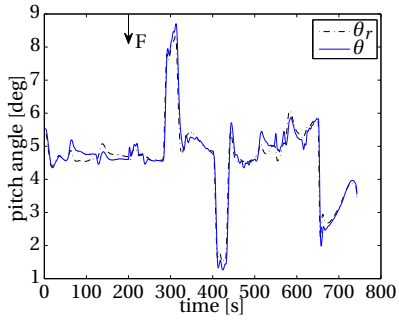


Figure 6.21: Pitch airspeed, engine separation.

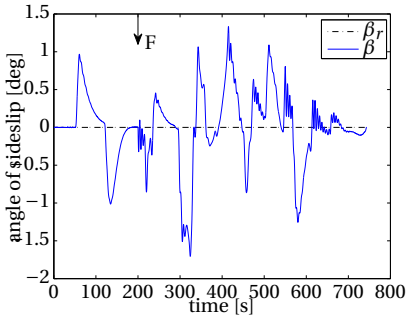


Figure 6.22: Angle of sideslip, engine separation.

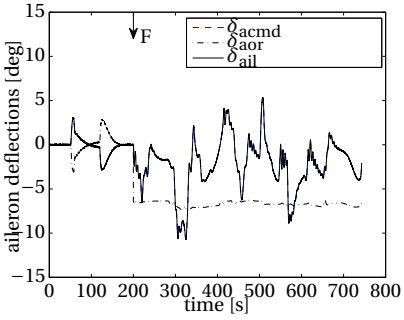


Figure 6.23: Commanded and actual aileron deflections, engine separation.

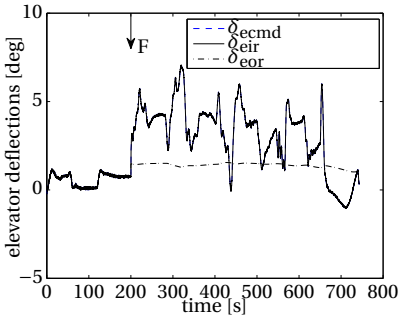


Figure 6.24: Commanded and actual elevator deflections, engine separation.

As shown in Figure 6.31, the rudder runaway failure occurs at the 200th second, and it produces a big influence on the body angular rates in a short transient period. The yawing rate r shows a spike around the 200th second. This is caused by two factors: the influence from the stuck rudder and the control reaction. Specifically, it is the rudder failure that makes the yawing rate r reaches -0.1 rad/s. While, it is the control reaction (using the differential thrust) as well as the side-force counteraction effect induced by the instantaneous nonzero β that makes r approach $+0.05$ rad/s.

The angular command tracking performance of the hybrid controller are illustrated in Figs. 6.33-6.35. ϕ and θ have nearly a zero tracking error, and β keeps smaller than 9.2 deg and it varies with time. The deterioration in the tracking performance of β is caused by the rudder runaway failure scenario, which nearly uses up all the control authority of the differential thrust. The non-zero tracking errors of β is comparable to those presented in [87] under the same fault scenario.

Figs. 6.36-6.37 provide the changing history of the commanded and actual (limited by the actuator dynamics and failures) control surface deflections for the roll and pitch channel respectively. In Figure 6.38, the actual control surface deflections of the stuck rudders are plotted. It should be noted that the deflection angles of the rudders vary although they are stuck to the left limit after the 200th second, see Figure 6.38. The

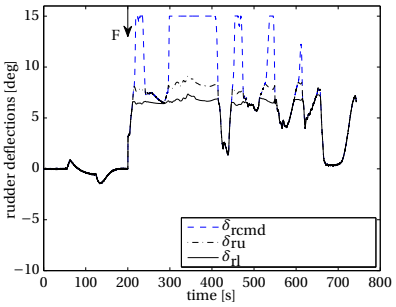


Figure 6.25: Commanded and actual rudder deflections, engine separation.

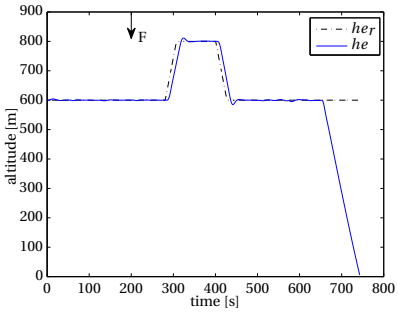


Figure 6.26: Altitude, engine separation.

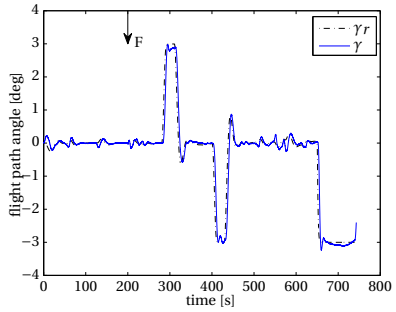


Figure 6.27: Flight path angle, engine separation.

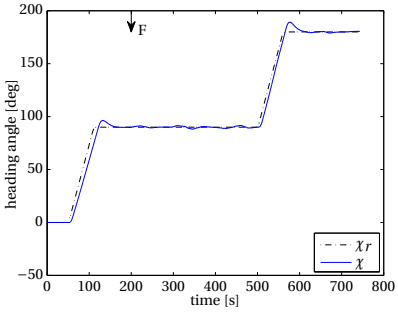


Figure 6.28: Heading angle, engine separation.

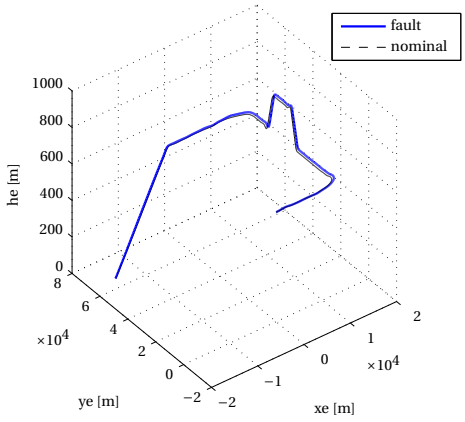


Figure 6.29: Three-dimensional trajectory, engine separation.

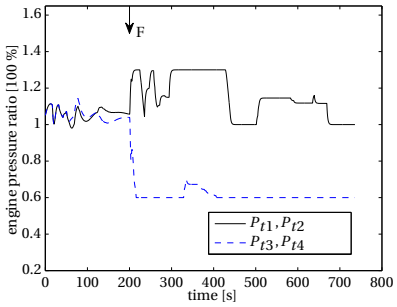


Figure 6.30: Engine Pressure Ratios, rudder runaway.

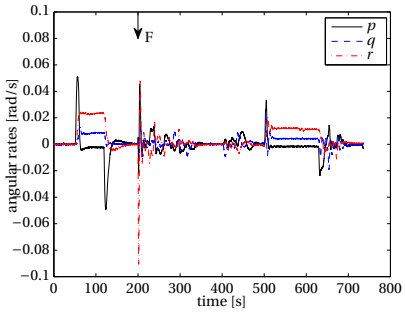


Figure 6.31: Angular rates, rudder runaway.

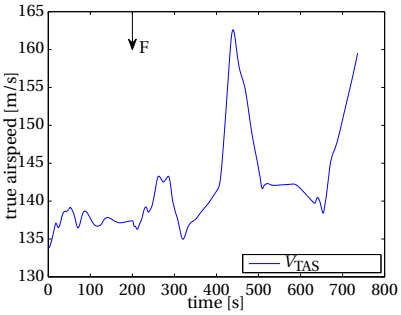


Figure 6.32: True airspeed, rudder runaway.

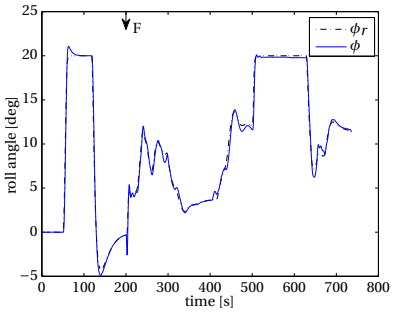


Figure 6.33: Roll angle, rudder runaway.

changes in deflection angles of rudders are caused by the fact that the aerodynamic blow-down has been taken into account by the RECOVER model. This has been confirmed by the fact that a correlation can be observed between (higher) airspeed and (lower) deflection angle, see [87].

Figs. 6.39-6.41 show that he, γ and χ are closely tracking their own reference command respectively. It also should be noted that the autopilot controller is switched from altitude controller into flight path hold controller at the 650^{th} second. That is, the aircraft is controlled in the altitude control mode during the first 650 seconds. Therefore, it is reasonable that the tracking error of γ appears to be relatively large in a few transient periods.

The changing history of the deflection angles of spoilers are depicted in Figure 6.42. The spoiler is assisting the ailerons to realize the control in the roll channel. Finally, the three-dimensional trajectory is shown in Figure 6.43. Again, the curve for the nominal case is borrowed from Figure 6.16. It can be observed from Figure 6.43 that a wider turn is needed by the post-failure aircraft. Though the roll angle is still kept around 20 deg, the sideslip angle is relatively big, which explains why the turn becomes wider.

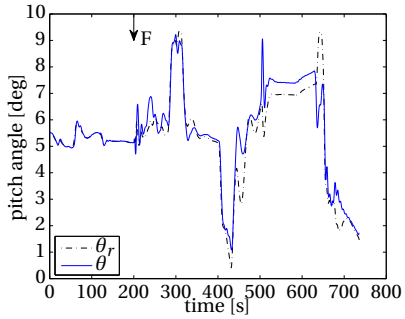


Figure 6.34: Pitch angle, rudder runaway.

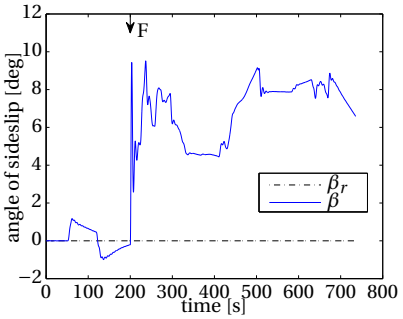


Figure 6.35: Angle of sideslip, rudder runaway.

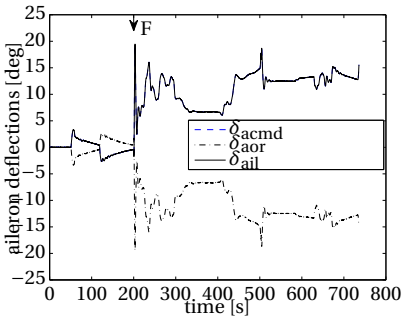


Figure 6.36: Commanded and actual aileron deflections, rudder runaway.

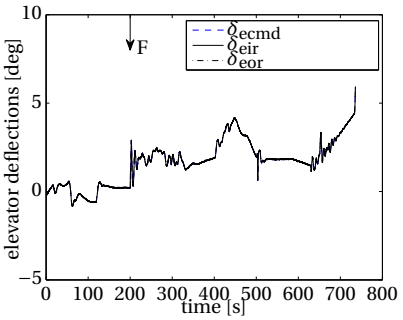


Figure 6.37: Commanded and actual elevator deflections, rudder runaway.

6.6. CONCLUSIONS

This paper has presented a new autopilot flight controller with four levels of control loops. The core of the autopilot flight controller is a hybrid NDI/SBB attitude control unit, which does not require real-time full aircraft model information. The controller is applied to the RECOVER model of the Boeing 747-200 aircraft, and evaluated using rudder runaway and EL AL 1862 benchmark fault scenarios developed by the GARTEUR FM-AG 16. The numerical simulation results show that the proposed hybrid NDI/SBB attitude controller can keep the safety of the aircraft even when the aforementioned failures occur, and can ensure a zero tracking error performance for roll angle and pitch angle commands as long as the aircraft is still controllable with the remaining valid control surfaces.

This paper uses the singular perturbation theory based sensor based backstepping (SBB) control approach, and extends its application to the body angular rate control of the Boeing 747-200 aircraft with special concern on sudden model changes. In addition, this SBB rate controller is combined with the NDI attitude controller and the PID flight path controller, and an autopilot flight controller has been synthesized. In the controller design, the control allocation problem is simplified by bounding a number of the control surfaces into a group. In addition, a second order command filter is adopted to enhance

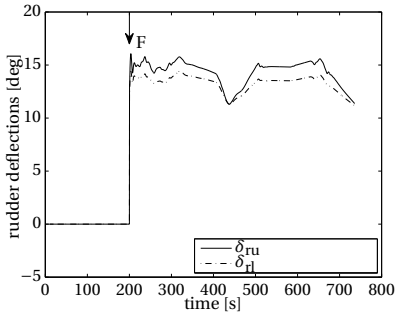


Figure 6.38: Actual rudder deflections, rudder runaway.

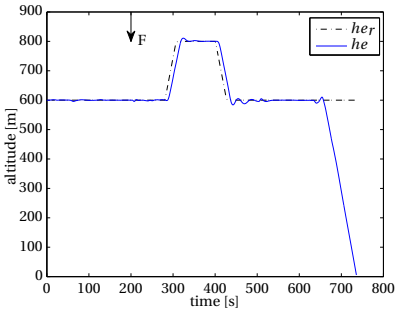


Figure 6.39: Altitude, rudder runaway.

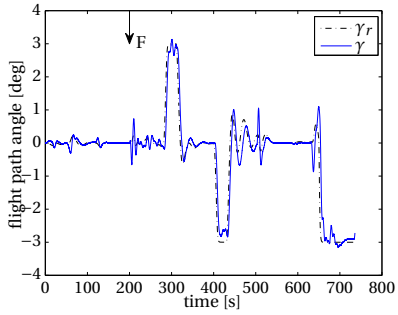


Figure 6.40: Flight path angle, rudder runaway.

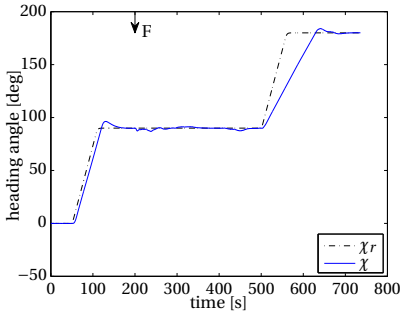


Figure 6.41: Heading angle, rudder runaway.

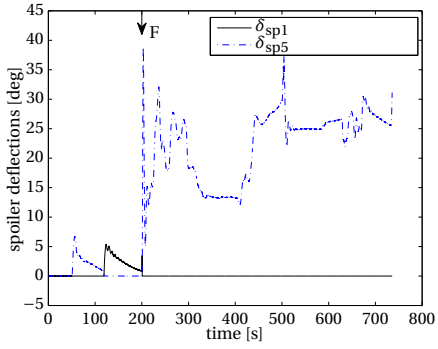


Figure 6.42: Actual spoiler deflections, rudder runaway.

the handling quality. Compared with the classic adaptive nonlinear dynamic inversion (ANDI) control approach or adaptive backstepping control law, the hybrid NDI/SBB attitude control setup needs less online model information. However, for the SBB body

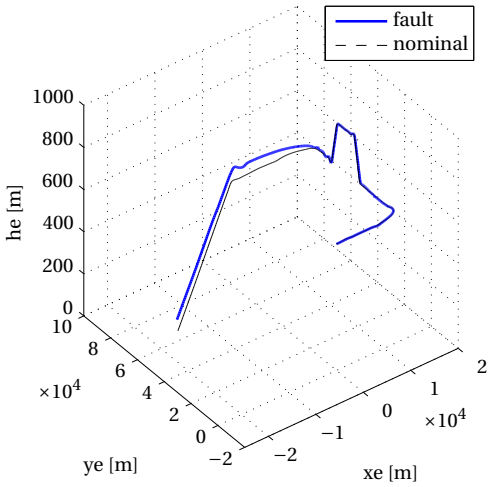


Figure 6.43: Three-dimensional trajectory, rudder runaway.

angular rate control approach, more research is needed into investigating the effects of time-delay in the actuator dynamics, as well as the measurement noise, before the method can be applied in real-world applications. For example, the influence from the engine response, which has a significant time-delay in real life (especially in low thrust levels), needs to be further investigated.

7

JOINT SENSOR BASED BACKSTEPPING APPROACH FOR FAULT-TOLERANT FLIGHT CONTROL

In Chapter 6, a hybrid double-loop angular controller was developed. Its inner-loop controller, i.e., the body angular rate controller, was designed using a sensor based backstepping control approach; its outer-loop controller, i.e., angular controller, was designed using a regular nonlinear dynamic inversion method. A potential drawback of this hybrid control structure is that two control loops are treated separately and the stability of the overall close-loop system is hard to prove. The Lyapunov function based recursive backstepping technique enables an overall controller to be designed loop by loop while guaranteeing the stability of the overall system in the Lyapunov sense. Therefore, we design a double-loop angular controller in this chapter using a recursive backstepping technique instead of a hybrid control structure. Similar to Chapter 6, the control power and adaptation ability of the new controller are validated using the RECOVER model against the engine separation failure and rudder runaway fault scenario.

The sensor based backstepping (SBB) control law, based on singular perturbation theory and Tikhonov's theory, is a novel incremental type high gain control approach. This Lyapunov function based method is not susceptible to model uncertainty since it uses measurements instead of onboard model variables. Considering these merits, we extend the SBB method to handle sudden structural changes in the fault tolerant flight control of a fixed wing aircraft. A new double-loop joint SBB attitude controller has been developed for a Boeing 747-200 aircraft using the backstepping technique. Compared with a double-loop nonlinear dynamic inversion angular control approach, the double-loop

SBB attitude control setup enables the verification of the system stability and allows more interaction between the angular rate loop and the angular loop. The benchmark rudder runaway and engine separation failure scenarios are employed to evaluate the proposed method. The simulation results show that the proposed joint SBB attitude control method can lead to a zero tracking error performance in the nominal condition and can lead to smaller sideslip angles than that led by the hybrid SBB approach when the aircraft is under failures.

7.1. INTRODUCTION

Under many post-failure circumstances, an aircraft can still achieve a certain level of flight performance with the remaining valid control effectors [88, 131]. However, as a consequence of the structural/actuator failures, the remaining control authority and the safe flight envelope of the aircraft is inevitably reduced.

Research on previous flight accidents [6, 131] shows that a suitable fault tolerant flight control (FTFC) approach is crucial to keep the stability and safety of the aircraft when structural failures or actuator failures occur. As the kernel of the FTFC system, a reconfigurable flight control algorithm needs to be powerful enough to tolerate the sudden changes occurring to an aircraft [6, 87, 131]. Possible candidates include robust control methods, e.g., sliding mode approach, model based adaptive nonlinear control approaches, e.g., adaptive nonlinear dynamic inversion and modular adaptive backstepping, and model-free nonlinear control methods, e.g., measurement based incremental nonlinear dynamic inversion and incremental backstepping [6, 119, 135, 163, 173, 174].

Recently, one promising fault tolerant control approach called adaptive nonlinear dynamic inversion (ANDI) was presented by Lombaerts et al. [87]. This method was applied to design a flight control system for the Reconfigurable Control for Vehicle Emergency Relief (RECOVER) benchmark model. To enable the close-loop control system to adapt to the post-failure state, a two-step online identification approach was employed to identify a physical aerodynamic model in real-time. However, the online model could become unreliable during the transient period after other more severe failures occur. In [87], Lombaerts et al. validated this ANDI controller in a piloted simulation on the Simulation, Motion, and Navigation (SIMONA) research simulator (SRS). Specifically, the rudder runaway case, the El Al flight 1862 fault, i.e., engine separation, and the stabilizer runaway scenarios were employed to evaluate the ANDI controller. The ANDI rate controller was proven to be able to enhance the stability of the post-failure aircraft and alleviate the high workload of the pilot.

Thereafter, Alwi and Edwards et al. [6] developed another promising reconfigurable flight controller using a model reference sliding mode control method with the assistance of a constant control allocation matrix. In [6], the sliding mode controller was also validated on the SRS using the El Al flight 1862 scenario. The sliding mode control scheme, which relies on relatively little information of the failure and the extent of the damage to the airframe, was also proven to be able to enhance the stability of the closed-loop system subject to a certain degree of model uncertainties (i.e.structural and actuator changes) caused by the separation of the right engine. However, the drawback of the sliding model control method is that the stability of the closed-loop system can be

only proven under several boundary assumptions concerning uncertainties. Similar to other robust control methods, the sliding mode controller, developed based on a state space model, only works in a limited neighborhood of a linearization point in theory.

In the past decade, nonlinear control approaches developed for singular perturbed systems have caught wide attention [67]. After combining a backstepping designing procedure with Tikhonov's Theorem and singular perturbation theory, Hovakimyan et al.[59] developed an approximation based nonlinear controller for non-affine systems. Later, this nonlinear control approach was extended by Falkena and van Oort et al. [45, 46] with the focus shifted to application. In [45, 46], the measurements of the derivatives of the controlled variables, e.g., body angular rates, are incorporated into the control solution. In addition, this controller indicated as the sensor based backstepping (SBB) approach was investigated against model uncertainties and measurement noises. In the SBB control approach, the Lyapunov function based backstepping design strategy guarantees multi-loop stability of the system [45, 59]. Actually, the SBB control approach should be viewed as a specific high-gain incremental nonlinear control scheme, which does not require online full model information. Unlike other incremental nonlinear control approaches (e.g.incremental NDI and incremental backstepping), an SBB control system does not require full knowledge of the control effectiveness matrix. Instead, it only needs to know the sign of the control effectiveness matrix.

More recently, Sun and de Visser et al. developed an angular controller using a hybrid sensor based backstepping (SBB) control approach in [139] and validated it using the RECOVER model. This hybrid SBB method consists of an angular controller designed using regular nonlinear dynamic inversion control laws and a body angular rate controller designed using the singular perturbation theory based SBB method from [46]. One advantage of the double-loop hybrid angular controller is that it can tolerate sudden changes induced by failures without requiring accurate online model information. Except for the control algorithm itself, [139] also presented a solution to the control allocation problem of a civil transportation aircraft, which matches the central control laws. In addition, differential thrust control was introduced to actively generate a yawing moment, which is required either by a flight task, e.g., a left turn, or by the need of counteracting the undesired yawing moment induced by a rudder failure or a vertical tail failure. The related simulation results are consistent with those indicated or presented in [23, 50, 158] and [48]. That is, differential thrust plays a crucial role in preventing an aircraft from loss-of-control in emergencies such as the rudder runaway case.

The objective of this chapter is to present an alternative fault-tolerant controller with its core a double-loop angular controller designed using a joint SBB control approach. The joint SBB control method presented in this chapter should be viewed as an extension of the hybrid SBB control approach presented in [139]. In [139], an angular controller, which is located in the second control loop, was designed using regular NDI control laws. As a consequence, the influences from body angles, i.e., ϕ, θ and β, are not directly involved in the body angular rate control. Most importantly, the stability of the double-loop hybrid NDI/SBB controller can not be guaranteed on a theoretical level. In this chapter, a joint SBB control method is designed using only a backstepping designing strategy, which greatly increases the interaction between the angular control loop and the body angular rate control loop. Most importantly, this Lyapunov function

based backstepping technique guarantees the stability of the overall double-loop control system. The joint SBB control approach is used to design an angular controller for a Boeing 747-200 aircraft model. The performance of the new joint SBB method is compared to that of the hybrid method in [139] for two different failure cases; the El Al flight 1862 scenario and the rudder runaway fault case. For the rudder runaway fault scenario, differential thrust provides the required yawing moment to counteract the asymmetric yawing moment induced by the failed rudders.

This chapter is outlined as follows. In Section 7.2, the overall control structure is introduced. Subsequently, the basic body angular rate and angular motion equations as well as the simplified version of a control allocation method are given by Section 7.3. Thereafter, a new double-loop angular controller based on the joint SBB approach is presented in Section 7.4. In Section 7.5, the performance of the new joint SBB method is compared to the hybrid SBB method from [139] in a number of simulations with the RECOVER model. Finally, Section 7.6 concludes this chapter.

7.2. VALIDATION PLATFORM

7.2.1. A VALIDATION BENCHMARK MODEL

The RECOVER benchmark model of Boeing 747-100/200 is a high-fidelity benchmark model developed for the verification of new FTFC techniques [131, 132]. Six benchmark fault scenarios have been embedded into this model. Since the El Al flight 1862 scenario and the rudder runaway fault case are the top two challenging cases [87], they are chosen in this chapter to validate the new joint SBB controller. The losses and the remaining functional control surfaces of the aircraft, when these two fault scenarios occur, have been summarized in [131, 139]. In the RECOVER benchmark model, the dynamics of EPR are modeled as $G(S) = \frac{1}{2 \cdot S + 1}$ with a changing rate limitation, i.e., 0.5 s^{-1}. It also should be mentioned that the engine response, in reality, has a significant lag especially when it is in low thrust level.

7.2.2. OVERALL AUTOPILOT FLIGHT CONTROL SYSTEM

Multi-loop designing setup using the SBB control approach has the advantage that it does not require real-time and accurate model information, e.g., the control effectiveness matrix. In addition, the backstepping multi-loop design plugs relatively more influence from the outer angular loop into the inner body angular rate loop, which is an effect of state feedbacks (compensation) from the outer loop.

As shown in Figure 7.1(a), an autopilot containing four control loops is designed for a Boeing 747-200 aircraft. Among these four loops, the altitude controller and flight path controller used in this chapter are exactly the same as those in [139]. In the flight path controller, $[\chi, \gamma, V]^\top$ are controlled variables. $[\phi, \theta, \beta]^\top$ are controlled variables in the angular control loop and $[p, q, r]^\top$ are the controlled variables in the first control loop. The airspeed is controlled by the collective engine pressure ratio (EPR) P_c. And differential EPRs (P_d) is used to actively generate a yawing moment. The spoilers are assisting the ailerons to generate adequate roll moment.

In [139], a hybrid SBB angular flight controller was designed after combining the nonlinear dynamic inversion technique, i.e., attitude controller, with the sensor based

backstepping technique, i.e., body rate controller. In this chapter, a joint double-loop angular controller will be designed for the Boeing 747-200 aircraft using only backstepping techniques.

In the nominal case or under the engine separation failure, the designed joint double-loop SBB angular controller uses regular functional control surfaces without introducing the differential thrust. The control structure is given by Figure 7.1(b) with $\mathbf{u} = [\delta_a, \delta_e, \delta_r]^\top$ the control input vector. On the contrary, the differential thrust is introduced in the joint double-loop SBB attitude controller to generate a required yawing moment to counteract the yawing moment induced by the rudder runaway fault when this failure occurs. The controller structure is depicted by Figure 7.1(c), where $\mathbf{u} = [\delta_a, \delta_e, P_d]^\top$.

In this chapter, the rudder runaway fault is assumed to be detectable. Once the rudder runaway fault is detected, the angular controller needs to be switched from the regular mode shown in Figure 7.1(b) to the differential thrust control mode shown in Figure 7.1(c). It should be noted that although differential thrust control has been widely investigated in the past few decades in the field of flight control, it is still a well known open issue that what is the best way to incorporate it into the regular flight control mode.

It needs to be noted that similar figures to Figure 7.1 have been given by [139]. However, they differ in the structure of the double-loop angular controller located in the central block.

To validate the new flight controller presented in the following sections, the simulated flight test benchmark depicted in Figure 7.2 is employed. Figure 7.2(a) shows the command sequences according to time axis. And Figure 7.2(b) describes a flight trajectory of a fault-free Boeing 747-200 aircraft equiped with a joint double-loop SBB angular controller.

All simulated flight tests in this chapter start from the same trim point.

7.3. EQUATIONS OF MOTION AND SIMPLIFIED RATE CONTROL ALLOCATION

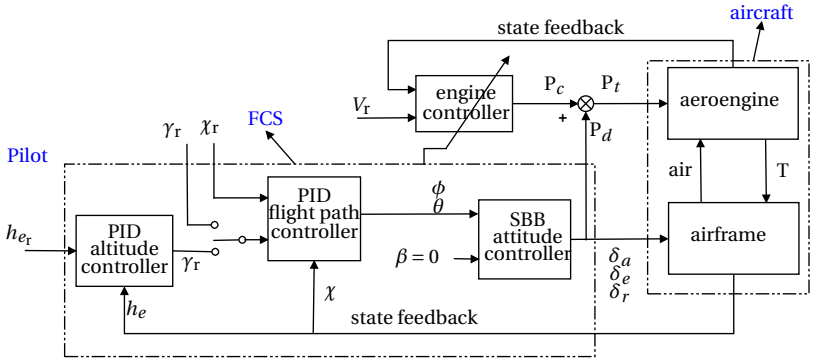
7.3.1. MOMENT AND ANGULAR RATE EQUATIONS

A Boeing 747-200 aircraft has 30 independent control inputs including 25 deflectable control surfaces, 4 engine pressure ratios (EPRs) and 1 flight gear (mode) input [131]. To simplify the control allocation logic, we can combine some of the aircraft inputs and use the following 19 equivalent control variables instead [5, 6, 158]:

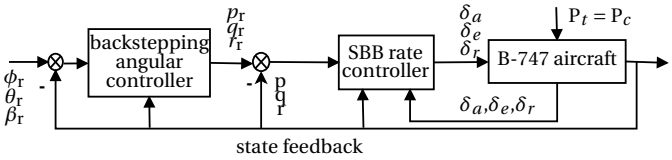
$$\mathbf{u} = \left[\delta_a, \delta_{sp}, \delta_e, \delta_{ih}, \delta_r, \mathbf{P}_t, P_d\right]^\top \tag{7.1}$$

where

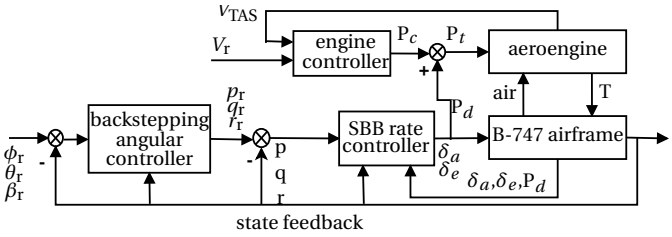
$$\begin{aligned}
\delta_a &= \left[\delta_{air}, \delta_{ail}, \delta_{aor}, \delta_{aol}\right] \\
\delta_{sp} &= \left[\left(\delta_{sp1} + \delta_{sp4}\right), \left(\delta_{sp2} + \delta_{sp3}\right), \left(\delta_{sp10} + \delta_{sp11}\right), \left(\delta_{sp9} + \delta_{sp12}\right)\right] \\
\delta_e &= \left[\delta_{eir}, \delta_{eil}, \delta_{eor}, \delta_{eol}\right] \\
\delta_r &= \left[\delta_{ru}, \delta_{rl}\right] \\
\mathbf{P}_t &= \left[P_{t_1}, P_{t_2}, P_{t_3}, P_{t_4}\right] \\
P_d &= \frac{1}{4}\left[\left(P_{t_1} - P_{t_4}\right) + \left(P_{t_2} - P_{t_3}\right)\right]
\end{aligned} \tag{7.2}$$



(a) Autopilot control architecture with four levels.



(b) Attitude controller, engine separation or nominal case.



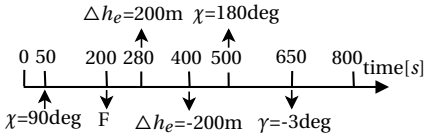
(c) Attitude controller, rudder runaway.

Figure 7.1: Fault-tolerant controller configuration.

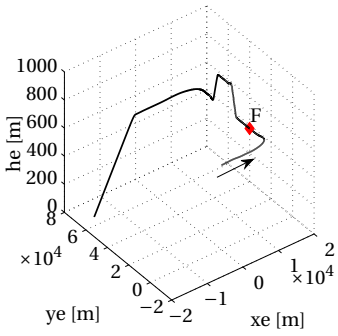
with P_c the collective engine pressure ratio (EPR), P_d the differential EPR and $\mathbf{P}_t$ the vector of total EPRs. They are defined as follows:

$$\begin{aligned} P_c &= mean(\mathbf{P}_t) \\ P_{t_1} &= P_{t_2} = P_c + P_d \\ P_{t_3} &= P_{t_4} = P_c - P_d \end{aligned} \tag{7.3}$$

The rotational equations of motion of the Boeing 747-200 aircraft have the following



(a) Simulated flight test benchmark.



(b) Three dimensional trajectory using joint SBB angular controller, nominal aircraft.

Figure 7.2: Designed command sequences for a flight test.

expression[88, 139]:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ \frac{w}{\sqrt{u^2+w^2}} & 0 & \frac{-u}{\sqrt{u^2+w^2}} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ A_\beta \end{bmatrix} \tag{7.4a}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = -\left\{ \mathbf{I}^{-1} \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right) - \frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1} \begin{bmatrix} bC_{l_{states}} \\ \bar{c}C_{m_{states}} \\ bC_{n_{states}} \end{bmatrix} \right\} + \frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1} \mathbf{M}_{\text{CA}} \cdot \mathbf{u} \tag{7.4b}$$

with

$$\mathbf{A}_\beta = \frac{1}{\sqrt{u^2+w^2}} \left[\frac{-uv}{V^2} \left(A_x - g\sin\theta \right) + \left(1 - \frac{v}{V^2} \right) \left(A_y + g\sin\phi\cos\theta \right) - \frac{vw}{V^2} \left(A_z + g\cos\phi\cos\theta \right) \right] \tag{7.5}$$

where A_x, A_y, A_z are the aerodynamic-related specific force components, i.e., the accelerations along the body axes without the gravitational effects, $\mathbf{M}_{\text{CA}}$ is the control allocation matrix. Note that, A_x, A_y, A_z are usually measured using accelerometers, the $\dot{\beta}$ equation in Eq. 7.4a is exact given accurate estimates of the translational variables, i.e., u, v, w. The aircraft dynamics in the angular rate and attitude levels constitute a standard second order cascaded system with the following expression:

$$\dot{\mathbf{x}}_1 = f(\mathbf{x}_1, \xi) + \mathbf{g}\mathbf{x}_2 \tag{7.6a}$$

$$\dot{\mathbf{x}}_2 = h(\mathbf{x}_1, \mathbf{x}_2, \xi) + \mathbf{k}\mathbf{u} \tag{7.6b}$$

with

$$\mathbf{x}_1 = \begin{bmatrix} \phi\, \theta\, \beta \end{bmatrix}^\top \tag{7.7a}$$

$$\mathbf{x}_2 = \begin{bmatrix} p\, q\, r \end{bmatrix}^\top \tag{7.7b}$$

$$f(\mathbf{x}_1, \xi) = \begin{bmatrix} 0\, 0\, \mathbf{A}_\beta \end{bmatrix}^\top \tag{7.7c}$$

$$\mathbf{g} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ \frac{w}{\sqrt{u^2+w^2}} & 0 & \frac{-u}{\sqrt{u^2+w^2}} \end{bmatrix} \tag{7.7d}$$

$$\mathbf{k} = \frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\mathbf{M}_{\mathrm{CA}} \tag{7.7e}$$

$$h(\mathbf{x}_1, \mathbf{x}_2, \xi) = -\mathbf{I}^{-1}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I}\begin{bmatrix} p \\ q \\ r \end{bmatrix}\right) + \frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\begin{bmatrix} bC_{l_{\mathrm{states}}} \\ \bar{c}C_{m_{\mathrm{states}}} \\ bC_{n_{\mathrm{states}}} \end{bmatrix} \tag{7.7f}$$

where $C_{l_{\mathrm{states}}}$, $C_{m_{\mathrm{states}}}$, $C_{n_{\mathrm{states}}}$ are the nondimensional moments contributed by all current states, the vector ξ denotes the system states excluding $\mathbf{x}_1, \mathbf{x}_2$, i.e., $\xi = [\alpha, u, v, w]^\top$. Keep in mind that the variables related to kinematic equations can always be known precisely. On the contrary, the variables bounded with body angular rate dynamics are always difficult to be accurately estimated due to model uncertainties. In the worst case, it is even not possible to obtain an accurate model which represents the body angular rate dynamics.

7.3.2. SIMPLIFIED CONTROL ALLOCATION

In order to introduce the control allocation problem more clearly, we take the control allocation problem in designing an adaptive NDI rate controller (see [87]) as an example. The control inputs can be calculated using the following formulation:

$$\mathbf{M}_{\mathrm{CA}} \cdot \mathbf{u} = \left\{ \frac{\mathbf{I}}{\frac{1}{2}\rho V^2 S}\left(\begin{bmatrix} \nu_p \\ \nu_q \\ \nu_r \end{bmatrix} + \mathbf{I}^{-1}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I}\begin{bmatrix} p \\ q \\ r \end{bmatrix}\right)\right) - \begin{bmatrix} bC_{l_{\mathrm{states}}} \\ \bar{c}C_{m_{\mathrm{states}}} \\ bC_{n_{\mathrm{states}}} \end{bmatrix} \right\} \tag{7.8}$$

with

$$\mathbf{M}_{\mathrm{CA}} = \begin{bmatrix} b & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & b \end{bmatrix} \mathbf{M}_E$$

where $\begin{bmatrix} \nu_p, \nu_q, \nu_r \end{bmatrix}^\top$ is the virtual body angular rate command vector, $\mathbf{M}_E$ is the control effectiveness matrix, $\mathbf{u}$ is the control input vector from Eq. 7.1.

In an adaptive NDI or classical adaptive backstepping rate controller, the unknown matrix $\mathbf{M}_E$ and the nondimensional moments induced by current states
, i.e., $C_{l_{\mathrm{states}}}, C_{m_{\mathrm{states}}}, C_{n_{\mathrm{states}}}$, need to be identified in real-time [87]. One representative aerodynamic model identification method is the two-step identification method [108].

Supposing that the matrix M_E in Eq. 7.8 has been identified, we can then design an optimizer to solve the control allocation problem described by Eq. 7.8. However, in order to purely demonstrate the power of the new control method proposed in this chapter, the M_E matrix used in this chapter is simplified in further:

$$M_E = \begin{bmatrix} \tilde{C}_{l\delta_a} & 0 & \tilde{C}_{l\delta_r} \\ 0 & \tilde{C}_{m\delta_e} & 0 \\ \tilde{C}_{n\delta_a} & 0 & \tilde{C}_{n\delta_r} \end{bmatrix} \tag{7.9}$$

with

$$\begin{aligned}
\tilde{C}_{l\delta_a} &= -C_{l\delta_{\text{air}}} + C_{l\delta_{\text{ail}}} - C_{l\delta_{\text{aor}}} + C_{l\delta_{\text{aol}}} - \\
&\quad C_{l\delta_{\text{sp1}}} - \cdots - C_{l\delta_{\text{sp4}}} + C_{l\delta_{\text{sp9}}} + \cdots + C_{l\delta_{\text{sp12}}} \\
\tilde{C}_{n\delta_a} &= -C_{n\delta_{air}} + C_{n\delta_{ail}} - C_{n\delta_{aor}} + C_{n\delta_{aol}} - \\
&\quad C_{n\delta_{\text{sp1}}} - \cdots - C_{n\delta_{\text{sp4}}} + C_{n\delta_{\text{sp9}}} + \cdots + C_{n\delta_{\text{sp12}}} \\
\tilde{C}_{m\delta_e} &= C_{m\delta_{\text{eir}}} + C_{m\delta_{\text{eil}}} + C_{m\delta_{\text{eor}}} + C_{m\delta_{\text{eol}}} \\
\tilde{C}_{l\delta_r} &= C_{l\delta_{\text{ru}}} + C_{l\delta_{\text{rl}}} \\
\tilde{C}_{n\delta_r} &= C_{n\delta_{\text{ru}}} + C_{n\delta_{\text{rl}}}
\end{aligned} \tag{7.10}$$

Due to this simplification, the control surfaces belonging to the same category would get equal deflecting commands.

7.4. JOINT ANGULAR/ANGULAR RATE CONTROLLER USING THE SBB APPROACH

The actuator dynamics can be viewed as a subsystem cascaded to the body angular rate dynamic system. Since the actuator dynamics are much faster than the body rate dynamics, an aircraft has the time-scale separation property. This allow us to use the SBB control approach to design a body angular rate controller for the Boeing 747-200 aircraft model. The structure of the double-loop angular controller are shown in Figure 7.1(a).

7.4.1. SBB CONTROL APPROACH

The control law designed using backstepping technique usually has a desirable property that the tracking error of the outer-loop commands are directly involved in regulating the inner loop control inputs. In order to fully explore this potential advantage in designing a multi-loop controller, a joint angular/angular rate controller is developed with two backstepping control loops using the SBB control method.

With regard to Eq. 7.6,the backstepping procedure starts by defining the tracking errors as:

$$\begin{cases} \mathbf{z}_1 = \mathbf{x}_1 - \mathbf{y}_r \\ \mathbf{z}_2 = \mathbf{x}_2 - \alpha \end{cases} \tag{7.11}$$

where α is the virtual control to be designed in the first step, and $\mathbf{y}_r = \begin{bmatrix} \phi_r & \theta_r & \beta_r \end{bmatrix}^\top$.
Step 1: Rewriting the $\mathbf{z}_1$ dynamics

$$\dot{\mathbf{z}}_1 = f(\mathbf{x}_1, \xi) + \mathbf{g}\mathbf{x}_2 - \dot{\mathbf{y}}_r = f(\mathbf{x}_1, \xi) + g(\alpha + \mathbf{z}_2) - \dot{\mathbf{y}}_r \tag{7.12}$$

We select a control Lyapunov function (CLF):

$$V_1(\mathbf{z}_1) = \frac{1}{2}\left[\mathbf{z}_1^\top \mathbf{z}_1 + \mathbf{k}_1 \lambda_1^\top \lambda_1\right] \tag{7.13}$$

where the gain $\mathbf{k}_1 > 0$ and the integrator term $\lambda_1 = \int_0^t \mathbf{z}_1 dt$ are introduced to eliminate the tracking error caused by the neglected control term, i.e., a term that is inevitably neglected due to un-modeled uncertainties. The derivative of V_1 is given by:

$$\dot{V}_1 = \mathbf{z}_1^\top \dot{\mathbf{z}}_1 + \mathbf{k}_1 \lambda_1^\top \mathbf{z}_1 = \mathbf{z}_1^\top \left[f(\mathbf{x}_1, \xi) + \mathbf{g}\mathbf{x}_2 - \dot{\mathbf{y}}_r + \mathbf{k}_1 \lambda_1\right] \tag{7.14}$$

The virtual control α is selected as:

$$\alpha = \mathbf{g}^{-1}\left[-\mathbf{c}_1\mathbf{z}_1 - f(\mathbf{x}_1,\xi) + \dot{\mathbf{y}}_r - \mathbf{k}_1\lambda_1\right] \tag{7.15}$$

to render the derivative

$$\dot{V}_1 = -\mathbf{c}_1\mathbf{z}_1^\top\mathbf{z}_1 \tag{7.16}$$

negative definite.
Step 2:
Rewriting the system in terms of the state $\mathbf{z_1}$ and $\mathbf{z_2}$:

$$\begin{cases}\dot{\mathbf{z}}_1 = f(\mathbf{x}_1,\xi) + \mathbf{g}(\alpha + \mathbf{z}_2) - \dot{\mathbf{y}}_r \\ \dot{\mathbf{z}}_2 = \dot{\mathbf{x}}_2 - \dot{\alpha} = h(\mathbf{x}_1,\mathbf{x}_2,\xi) + \mathbf{k}\mathbf{u} - \dot{\alpha}\end{cases} \tag{7.17}$$

The CLF in Eq. 7.13 is augmented for the $(\mathbf{z}_1,\mathbf{z}_2)$-system with an extra term that penalizes the tracking error $\mathbf{z}_2$:

$$V_2(\mathbf{z}_1,\mathbf{z}_2) = \frac{1}{2}\mathbf{z}_1^\top\mathbf{z}_1 + \frac{1}{2}\mathbf{k}_1\lambda_1^\top\lambda_1 + \frac{1}{2}\mathbf{z}_2^\top\mathbf{z}_2 + \frac{1}{2}\mathbf{k}_2\lambda_2^\top\lambda_2 \tag{7.18}$$

Taking the derivative of V_2 results in

$$\begin{aligned}\dot{V}_2 &= \mathbf{z}_1^\top\dot{\mathbf{z}}_1 + \mathbf{k}_1\lambda_1^\top\mathbf{z}_1 + \mathbf{z}_2^\top\dot{\mathbf{z}}_2 + \mathbf{k}_2\lambda_2^\top\mathbf{z}_2 \\ &= \mathbf{z}_1^\top\left\{f(\mathbf{x}_1,\xi) + \mathbf{g}\left[\mathbf{g}^{-1}\left(-\mathbf{c}_1\mathbf{z}_1 - f(\mathbf{x}_1,\xi) + \dot{\mathbf{y}}_r - \mathbf{k}_1\lambda_1\right) + \mathbf{z}_2\right] - \dot{\mathbf{y}}_r\right\} \\ &+ \mathbf{k}_1\lambda_1^\top\mathbf{z}_1 + \mathbf{z}_2^\top\left(h(\mathbf{x}_1,\mathbf{x}_2,\xi) + \mathbf{k}\mathbf{u} - \dot{\alpha}\right) + \mathbf{k}_2\lambda_2^\top\mathbf{z}_2 \\ &= -\mathbf{c}_1\mathbf{z}_1^\top\mathbf{z}_1 + \mathbf{z}_2^\top\left(\mathbf{g}\mathbf{z}_1 + h(\mathbf{x}_1,\mathbf{x}_2,\xi) + \mathbf{k}_2\lambda_2 + \mathbf{k}\mathbf{u} - \dot{\alpha}\right)\end{aligned} \tag{7.19}$$

Then we can get a regular backstepping control law:

$$\mathbf{u} = \mathbf{k}^{-1}\left(-\mathbf{c}_2\mathbf{z}_2 - \mathbf{g}\mathbf{z}_1 + \dot{\alpha} - h(\mathbf{x}_2) - \mathbf{k}_2\lambda_2\right) \tag{7.20}$$

To design a sensor based backstepping (SBB) controller, we do not need $h(\mathbf{x}_1,\mathbf{x}_2,\xi)$ and $\mathbf{k}$ which represent the dynamics of the angular rate loop. Instead of substituting $\dot{\mathbf{z}}_2 = \dot{\mathbf{x}}_2 - \dot{\alpha} = h(\mathbf{x}_1,\mathbf{x}_2,\xi) + \mathbf{k}\mathbf{u} - \dot{\alpha}$ in Eq. 7.19, $\dot{\mathbf{z}}_2$ is provisionally viewed as control inputs:

$$\begin{aligned}\dot{V}_2 &= \mathbf{z}_1^\top\dot{\mathbf{z}}_1 + \mathbf{k}_1\lambda_1^\top\mathbf{z}_1 + \mathbf{z}_2^\top\dot{\mathbf{z}}_2 + \mathbf{k}_2\lambda_2^\top\mathbf{z}_2 \\ &= \mathbf{z}_1^\top\left\{f(\mathbf{x}_1) + \mathbf{g}\left[\mathbf{g}^{-1}\left(-\mathbf{c}_1\mathbf{z}_1 - f(\mathbf{x}_1) + \dot{\mathbf{y}}_r - \mathbf{k}_1\lambda_1\right) + \mathbf{z}_2\right] - \dot{\mathbf{y}}_r\right\} \\ &+ \mathbf{k}_1\lambda_1^\top\mathbf{z}_1 + \mathbf{z}_2^\top\dot{\mathbf{z}}_{2\text{des}} + \mathbf{k}_2\lambda_2^\top\mathbf{z}_2 \\ &= -\mathbf{c}_1\mathbf{z}_1^\top\mathbf{z}_1 + \mathbf{z}_2^\top\left(\mathbf{g}\mathbf{z}_1 + \mathbf{k}_2\lambda_2 + \dot{\mathbf{z}}_{2\text{des}}\right)\end{aligned} \tag{7.21}$$

In order to make $\dot{V}_2$ negative definite, we can select:

$$\dot{\mathbf{z}}_{2\text{des}} = -\mathbf{c}_2\mathbf{z}_2 - \mathbf{k}_2\lambda_2 - \mathbf{g}\mathbf{z}_1 \tag{7.22}$$

Following Eq. 7.4b, an equivalent control input vector $\mathbf{u}_{\text{red}}$ is defined as:

$$\mathbf{u}_{\text{red}} = \mathbf{M}_{\text{CA}}\cdot\mathbf{u} \tag{7.23}$$

It should be noted that $\mathbf{M}_{\text{CA}}$ is assumed to be time invariant in Eq. 7.23 in this chapter.

After substituting Eq. 7.23 into Eq. 7.4b, the SBB controller for Eq. 7.4 can be designed as follows [59]:

$$\begin{aligned}\epsilon\dot{\mathbf{u}}_{\text{red}} &= -sgn\left(\frac{\partial\dot{\mathbf{z}}_2}{\partial\mathbf{u}_{\text{red}}}\right)\cdot[\dot{\mathbf{z}}_2-\dot{\mathbf{z}}_{2\text{des}}]\\ &= -sgn\left(\frac{\partial\dot{\mathbf{z}}_2}{\partial\mathbf{u}_{\text{red}}}\right)\cdot[\dot{\mathbf{z}}_2+\mathbf{g}\mathbf{z}_1+\mathbf{k}_2\lambda_2+\mathbf{c}_2\mathbf{z}_2]\end{aligned} \tag{7.24}$$

where ϵ is a tuning parameter with small positive value, i.e., $0<\epsilon<<1$, $\dot{\mathbf{z}}_2=\dot{\mathbf{x}}_2-\dot{\alpha}$, $\dot{\mathbf{x}}_2$ is measurable and $\dot{\alpha}$ can be calculated according to Eq. 7.15.

According to Eq. 7.11 and Eq. 7.4b, we have

$$\frac{\partial\dot{\mathbf{z}}_2}{\partial\mathbf{u}_{\text{red}}}=\frac{1}{2}\rho V^2 S\cdot\mathbf{I}^{-1} \tag{7.25}$$

Therefore, the controller in Eq. 7.24 becomes:

$$\dot{\mathbf{u}}_{\text{red}}=-\frac{1}{\epsilon}\cdot sgn\left(\frac{1}{2}\rho V^2 S\cdot\mathbf{I}^{-1}\right)\cdot[\dot{\mathbf{z}}_2+\mathbf{g}\mathbf{z}_1+\mathbf{k}_2\lambda_2+\mathbf{c}_2\mathbf{z}_2] \tag{7.26}$$

The dynamic pressure is assumed constant during our failure cases. Considering the variation of dynamic pressure, the reader is referred to retune ϵ in Eq. 7.26, but this topic is out of the scope of this thesis. By integrating $\dot{\mathbf{u}}_{\text{red}}$, the equivalent control inputs $\mathbf{u}_{\text{red}}$ can be derived:

$$\mathbf{u}_{\text{red}_k}=\mathbf{u}_{\text{red}_{k-1}}+\int_{(k-1)\text{T}}^{k\text{T}}\dot{\mathbf{u}}_{\text{red}}\cdot dt \tag{7.27}$$

According to Eq. 7.23, the control input $\mathbf{u}$ can be solved using a control allocation algorithm if $\mathbf{M}_{\text{CA}}$ from Eq. 7.8 is available. Unfortunately, the need of $\mathbf{M}_{\text{CA}}$ makes the controller dependent on the aerodynamic model. It is assumed in this chapter that a fixed $\mathbf{M}_{\text{CA}}$ is available around a trim point.

To simplify the implementation of the controller, the requirement of control allocation is removed from the design procedure. Remember that the control surfaces have been categorized into 3 groups (see. Eq. 7.10). From each group, a control surface is chosen as the representative control input. Consequently, a representative control input vector $\mathbf{u}_{\text{rep}}=[\delta_{1_{\text{rep}}},\delta_{2_{\text{rep}}},\delta_{3_{\text{rep}}}]$ can be derived. For example, the representatives can be selected as follows: $\delta_{1_{\text{rep}}}=\delta_{\text{ail}}$, $\delta_{2_{\text{rep}}}=\delta_{\text{eil}}$ and $\delta_{3_{\text{rep}}}=\delta_{\text{ru}}$ (or $\delta_{3_{\text{rep}}}=P_d$). The control allocation matrix $\mathbf{M}_{\text{CA}_{\text{rep}}}$ can be calculated from the aforementioned fixed matrix $\mathbf{M}_{\text{CA}}$ using Eq. 7.9. It should be noted that the following assumption is made at this place:

$$\mathbf{M}_{\text{CA}_{\text{rep}}}\cdot\mathbf{u}_{\text{rep}}\approx\mathbf{M}_{\text{CA}}\cdot\mathbf{u} \tag{7.28}$$

After substituting Eq. 7.28 into Eq. 7.6b, the SBB controller can be redesigned for the system Eq. 7.6. Consequently, the term $\frac{\partial\dot{\mathbf{z}}_2}{\partial\mathbf{u}_{\text{red}}}$ in Eq. 7.24 and Eq. 7.25 should be replaced by $\frac{\partial\dot{\mathbf{z}}_2}{\partial\mathbf{u}_{\text{rep}}}$, and Eq. 7.25 becomes:

$$-sgn\left(\frac{\partial\dot{\mathbf{z}}_2}{\partial\mathbf{u}_{\text{rep}}}\right)=-sgn\left(\frac{1}{2}\rho V^2 S\cdot\mathbf{I}^{-1}\mathbf{M}_{\text{CA}_{\text{rep}}}\right) \tag{7.29}$$

Correspondingly, Eq. 7.26 becomes:

$$\dot{\mathbf{u}}_{\text{rep}} = -\frac{1}{\epsilon} sgn\left(\frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\mathbf{M}_{\text{CA}_{\text{rep}}}\right) \cdot \left[\dot{\mathbf{z}}_2 + \mathbf{g}\mathbf{z}_1 + \mathbf{k}_2\lambda_2 + \mathbf{c}_2\mathbf{z}_2\right] \tag{7.30}$$

Subsequently, $\mathbf{u}_{\text{rep}}$ can be calculated using Eq. 7.27. In calculating $sgn\left(\frac{1}{2}\rho V^2 S \cdot \mathbf{I}^{-1}\mathbf{M}_{\text{CA}_{\text{rep}}}\right)$, the same method as that used in [59] is adopted. That is, only the sign of the diagonal elements of the matrix (in the bracket) are used in designing the controller.

The configuration of the double-loop attitude controller has the framework shown in Figure 7.3. In this figure, 'TA' is the acronym of a tuning algorithm, and τ represents the

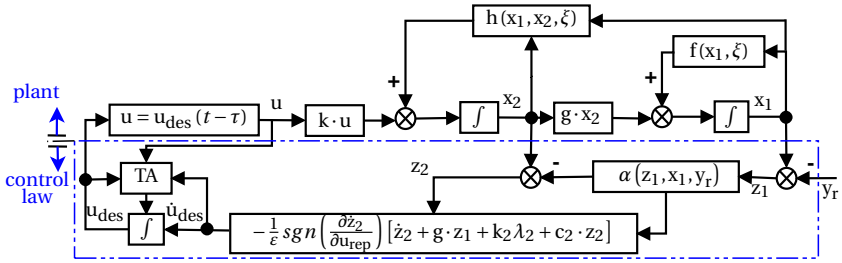


Figure 7.3: Flow chart of the joint SBB attitude controller.

time delay between the commanded and the actual control inputs. Detailed description about 'TA' block can be found in [139]. As can be seen from Figure 7.3, the effects from the outer loop dynamics denoted by the term $\mathbf{g}\mathbf{z}_1$ are directly involved in regulating the control inputs in the inner loop.

7.4.2. INCREMENTAL BACKSTEPPING, SENSOR BASED BACKSTEPPING AND THEIR RELATION

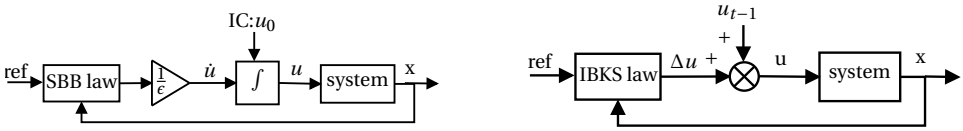
For a two-loop angular/angular rate controller designed for a system described by Eq. 7.6, the regular IBKS and the singular perturbation based IBKS, i.e., the sensor based backstepping approach, have the following forms:

$$\triangle\mathbf{u} = -\mathbf{k}^{-1} \cdot \left(\mathbf{c}_2\mathbf{z}_2 + [\dot{\mathbf{x}}_2 - \dot{\mathbf{x}}_{2r}] + \mathbf{g}\mathbf{z}_1\right) \tag{7.31}$$

$$\dot{\mathbf{u}} = -\frac{1}{\epsilon} sign\left(\frac{\partial \dot{z}_2}{\partial u}\right)\left(\mathbf{c}_2\mathbf{z}_2 + [\dot{\mathbf{x}}_2 - \dot{\mathbf{x}}_{2r}] + \mathbf{g}\mathbf{z}_1\right) \tag{7.32}$$

The reader is referred to [2, 44] for more details on designing a two-loop regular incremental backstepping controller. The control diagrams for both control methods are given in Figure 7.4.

Comparing Eq. 7.32 with Eq. 7.31, it can be noticed that the two approaches are theoretically similar in the sense that both of them employ one-step control adjustment, i.e., Δu or $\int_{(k-1)\text{T}}^{k\text{T}} \dot{\mathbf{u}} \cdot dt$. More precisely, the following links between these two controllers



(a) SBB control diagram.

(b) IBKS control diagram.

Figure 7.4: Block diagrams for SBB and IBKS control systems.

exist:

$$\triangle \mathbf{u} = \int_{(k-1)\mathrm{T}}^{k\mathrm{T}} \dot{\mathbf{u}} \cdot dt \triangleq \dot{\mathbf{u}} \cdot T \tag{7.33}$$

if and only if the absolute values of the diagonal elements in $\mathbf{k}^{-1}$ equal to $\frac{1}{\epsilon}$, where T, ϵ are the sampling time and the time scale parameter. Regarding the relationship between the SBB and IBKS approaches, the ϵ should be understood as a parameter determined by control derivatives contained by the $\mathbf{k}$ matrix, see Eq. 7.6, and the sampling time T. The SBB approach can be viewed as a special case for an acceleration mesurement-based incremental backstepping control method since they both perform only one-step incremental regulation at each sampling time instant.

7.4.3. COMMAND FILTER AND DIFFERENTIATOR

A command filter is needed to regulate the given reference commands to enable that the closed-loop aircraft system has adequate handling qualities. In addition, the command filter is also playing a crucial role in constraining the aircraft within the safe flight envelope. The reference rate commands should be scheduled according to aircraft speed, altitude and damage degree.

The saturation effects from the integrator need to be removed. When some failures happen to the control surfaces or the control effector such as the propulsion system has a big time delay, these saturation effects become much more serious. In this paper, a tuning algorithm (TA) block was designed to handle this by evaluating the discrepancy information between the measured control inputs $\mathbf{u}_m$ and the integrator outputs $\mathbf{u}_{\mathrm{des}}$, see Figure 7.3. The mechanism of the 'TA' block is described as follows:

$$\mathbf{u}_{\mathrm{des}} = \int_{0}^{k\mathrm{T}} \dot{\mathbf{u}}_{\int} \cdot dt \tag{7.34}$$

with

$$\dot{\mathbf{u}}_{\int} = \begin{cases} 0, & \text{if } \left(\mathbf{u}_{\mathrm{u}} - \mathbf{u}_{\mathrm{des}}\right) \leq 0 \text{ and } \dot{\mathbf{u}}_{\mathrm{des}} \leq 0; \text{ or } \left(\mathbf{u}_{\mathrm{des}} - \mathbf{u}_{\mathrm{l}}\right) \leq 0 \text{ and } \dot{\mathbf{u}}_{\mathrm{des}} \geq 0; \text{ or } t\left(\left|\mathbf{u}_{\mathrm{m}} - \mathbf{u}_{\mathrm{des}}\right| > \varsigma_1\right) > \varsigma_2 \\ \dot{\mathbf{u}}_{\mathrm{des}}, & else \end{cases} \tag{7.35}$$

where $\mathbf{u}_{\mathrm{des}}$ and $\mathbf{u}_m$ are the controller output and measured control input respectively, and $\mathbf{u}_{\mathrm{l}}$ and $\mathbf{u}_{\mathrm{u}}$ are the lower and upper position limits. The ς_1 and ς_2 are constant values acting as thresholds which depend on the dynamics or time delay of the actuators. The $t(\cdot)$ function calculates the lasting time for the case where the conditions in the bracket are satisfied. It can be noticed from Eq. 7.35 that the first two conditions in the 'if' branch

Table 7.1: Joint SBB attitude controller parameters, $\epsilon = 0.15$, nominal/engine separation

Channel	proportion	integration
Angular control	[1, 0.5, 2]	[0, 0, 0]
Body rate control	[0.1, 0.2, 0.1]	[0, 0, 0]

aim at removing integrator saturation, the third condition takes into account long-term actuator time delay or actuator failures, e.g., control surfaces that are stuck.

In this chapter, we assume that angular accelometers are not available. A 5th-order sliding mode differentiator presented in [78] is employed to calculate high quality $[\dot{p},\ \dot{q},\ \dot{r}]^\top$ using filtered signals of $[p,\ q,\ r]^\top$, which are outputs of an extended Kalman filter block.

7.5. RESULTS AND ANALYSIS

To validate the joint double-loop SSB angular controller developed in Section 7.4, a flight path controller, see Figure 7.1(a), designed using regular PID control laws are combined with the angular controller forming a new autopilot flight controller. In addition, an independent propulsion controller was designed for the aeroengine using PID. In this engine controller, P_c and V_{TAS} are the control input and controlled variable respectively.

The new autopilot flight controller will be applied to the Boeing 747-200 aircraft. It is firstly evaluated for the nominal case and then evaluated for two fault scenarios: rudder runaway and right engine separation. Eq. 7.10 and Eq. 7.3 were adopted for control allocation purpose.

7.5.1. COMMAND FILTER SETUP AND ACTUATOR WORKING RANGE

The actuators of the control surfaces are modeled as first order low-pass filters. Their saturation limits and deflection rate limits have been presented in [139]. In addition, a command filter presented in [47] is utilized in this chapter. This filter has an adjustable natural frequency ω_n and damping ratio ζ. For the scheduling limits on the body angular rate commands and the attitude angular commands, we use the same setups as those presented in [139].

7.5.2. VALIDATION RESULTS OF THE NOMINAL AIRCRAFT

The proposed double-loop SBB angular controller (see Figure 7.1(b)) is evaluated by flying the Boeing 747-200 aircraft in the nominal (fault-free) state. In addition, the simulation results derived using the joint SBB method are compared with those obtained using the hybrid SBB method (see ref.[139]). The idea of a fault-free test of the controller is to show the nominal capability of the proposed controller. The chosen controller parameters are listed in Table 7.1. It should be mentioned that the autopilot presented in this chapter uses the same flight path controller and altitude controller as those in [139].

Figures 7.5-7.19 illustrate the validation results of the designed controller under the nominal case. Outside the autopilot, the reference command sequences shown in Figure 7.2(a) were performed. Figure 7.5 shows the changing history of EPRs, and

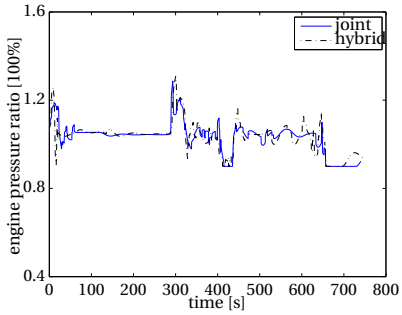


Figure 7.5: Engine Pressure Ratios, nominal.

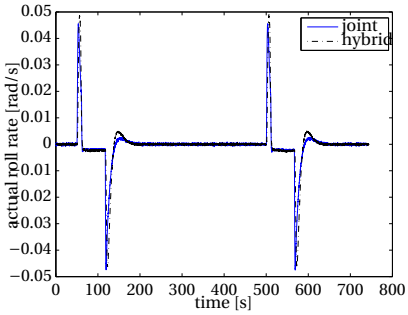


Figure 7.6: Roll rates, nominal.

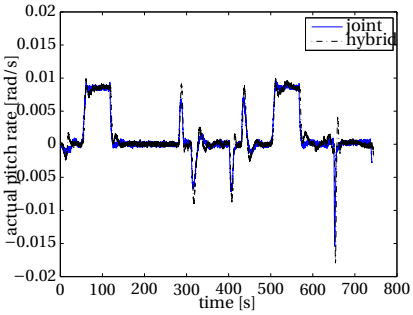


Figure 7.7: Pitch rates, nominal.

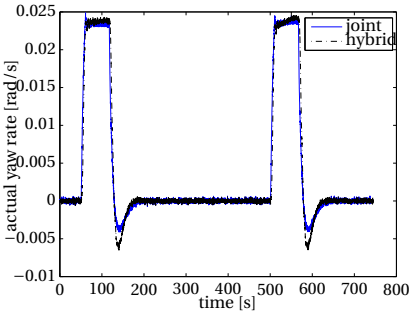


Figure 7.8: Yaw rates, nominal.

7

the EPRs are regulated to keep the true airspeed between 122 m/s and 142 m/s, see Figure 7.9. In Figs. 7.6-7.8, the changes in body angular rates are illustrated. During most of the time period, the curve for the joint SBB method and that of the hybrid SBB method overlap. However, the body rate changes for the hybrid SBB method are slightly more aggressive than those for the joint SBB method during transient periods. The angular tracking performance is depicted in Figs. 7.10-7.12. The inner angular controller using the joint SBB approach enables the aircraft to closely track the angular reference commands. Compared with the hybrid SBB approach, the joint SBB method leads to slightly more moderate changes of the roll angle and pitch angle. In addition, the joint method leads to a much smaller β than that of the hybrid SBB method according to Figure 7.12.

Figs. 7.13-7.15 show the commanded and actual control deflections of ailerons, elevators and rudders respectively. As can be seen, the actual control surface deflections match the commanded ones due to the fact that the aircraft is fault-free. Compared with the hybrid SBB method, the joint SBB method leads to more desired control deflections of ailerons and rudders, see Figure 7.13 and Figure 7.15. This is the reason that the joint SBB method leads to a much smaller β, see Figure 7.12, from 50 s to 100 s.

Finally, the command tracking performance in the flight path control level is demonstrated in Figs. 7.16-7.18. It can be clearly seen that both control methods can lead

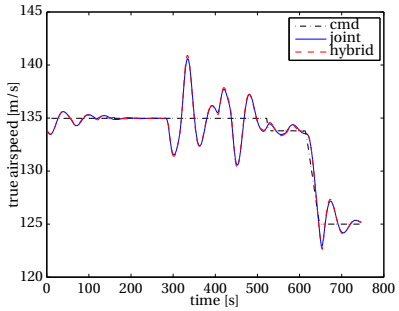


Figure 7.9: True airspeed, nominal.

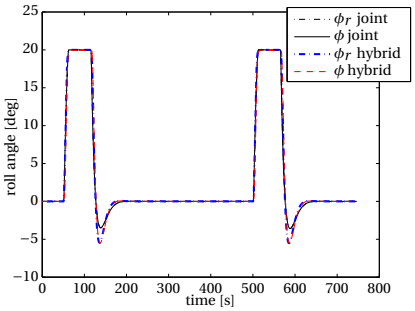


Figure 7.10: Roll angle, nominal.

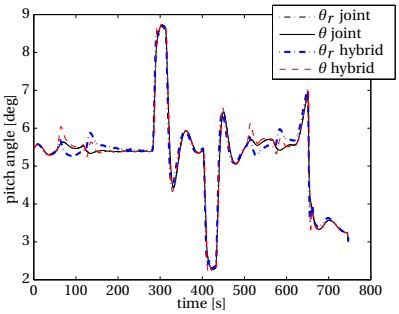


Figure 7.11: Pitch angle, nominal.

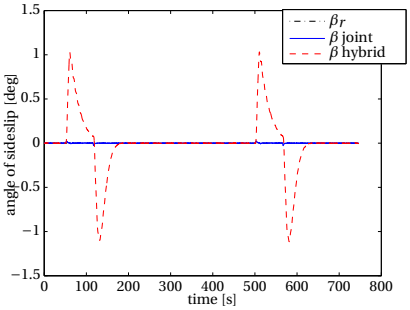


Figure 7.12: Angle of sideslip, nominal.

to a relatively good tracking performance in the χ, γ and he channel respectively. To add visualization clarity of the flight task, a three-dimensional trajectory of the aircraft is plotted in Figure 7.19. Figure 7.19(a) shows two three-dimensional trajectories of an aircraft which are obtained using the joint controller and hybrid controller respectively. To enhance the clarity, the corresponding ground tracks are given by Figure 7.19(b). A conclusion can be drawn from Figure 7.19 that an autopilot using the joint SBB method leads to a quite similar flight path to that led by an autopilot using the hybrid SBB control approach. The tiny difference between those two trajectories are caused by the difference of airspeed.

7.5.3. VALIDATION RESULTS UNDER TWO BENCHMARK FAILURE SCENARIOS

In the first numerical simulation experiment, the joint SBB attitude controller shown in Figure 7.1(b) is validated by flying the aircraft under the engine separation failure scenario. This controller parameters are tabulated in Table 7.1. The validation results are given by Figs. 7.20-7.32.

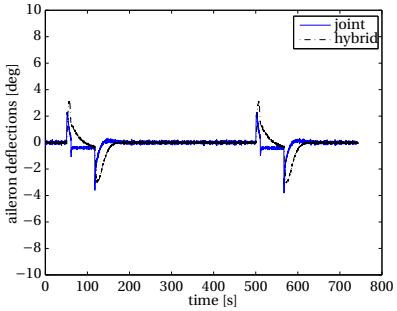


Figure 7.13: Commanded and actual aileron deflections, nominal.

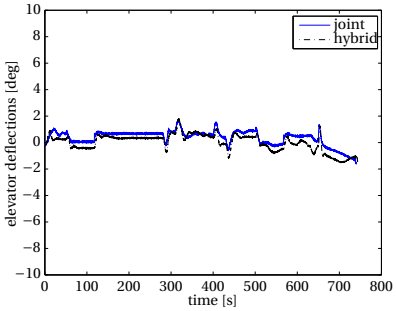


Figure 7.14: Commanded and actual elevator deflections, nominal.

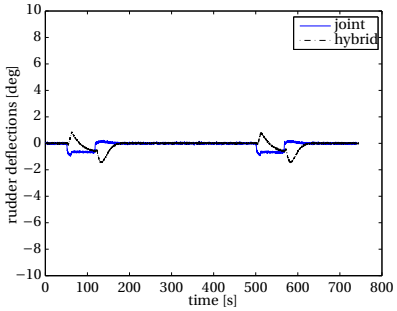


Figure 7.15: Commanded and actual rudder deflections, nominal.

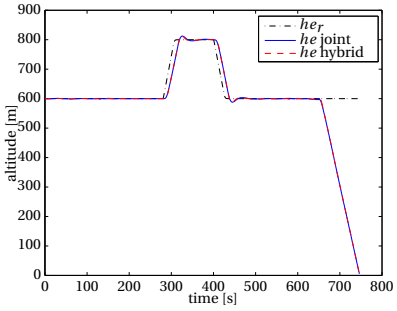


Figure 7.16: Altitude, nominal.

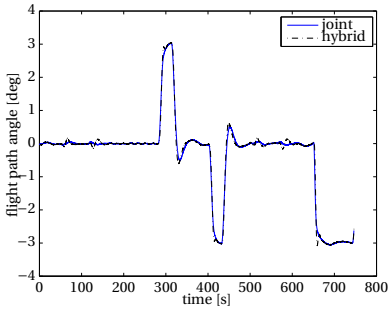


Figure 7.17: Flight path angle, nominal.

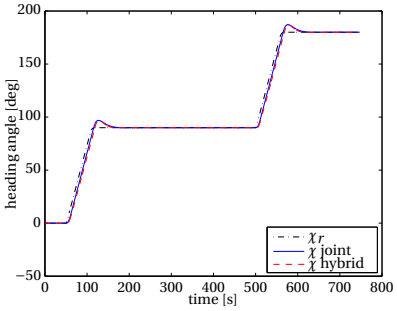
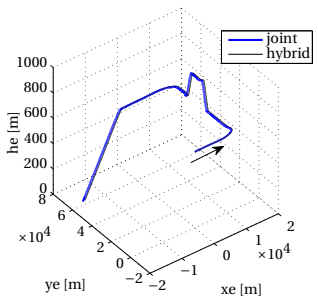
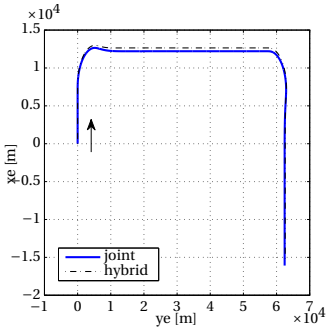


Figure 7.18: Heading angle, nominal.



(a) Three-dimensional trajectory, nominal.



(b) Ground track, nominal.

Figure 7.19: Trajectory comparison results using the joint and hybrid SBB angular controllers respectively.

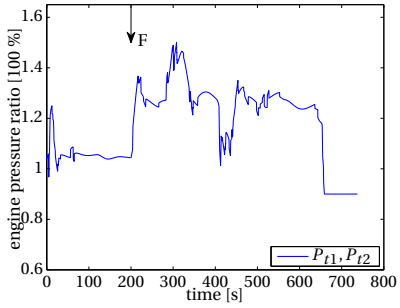


Figure 7.20: Engine Pressure Ratios, engine separation.

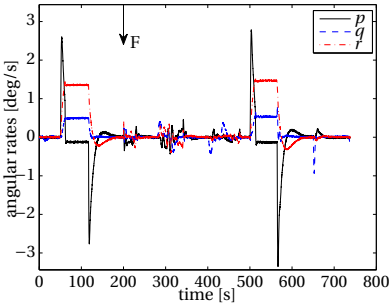


Figure 7.21: Angular rates, engine separation.

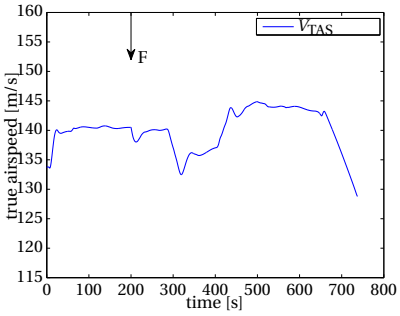


Figure 7.22: True airspeed, engine separation.

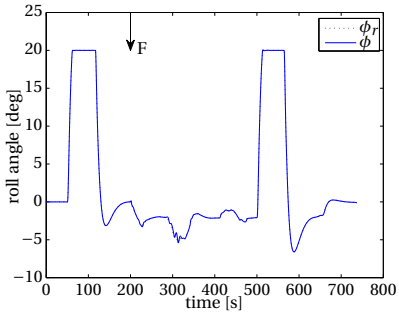


Figure 7.23: Roll angle, engine separation.

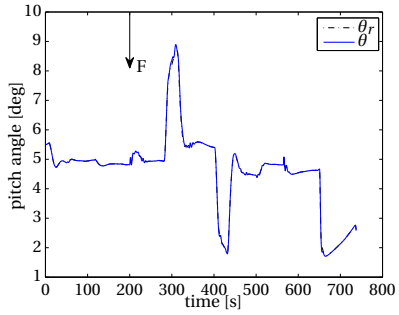


Figure 7.24: Pitch angle, engine separation.

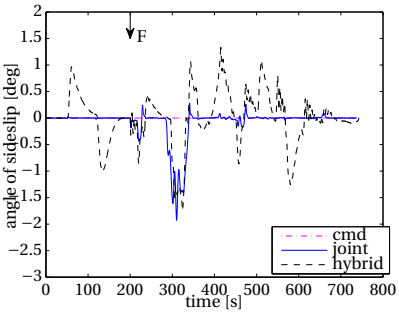


Figure 7.25: Angle of sideslip, engine separation.

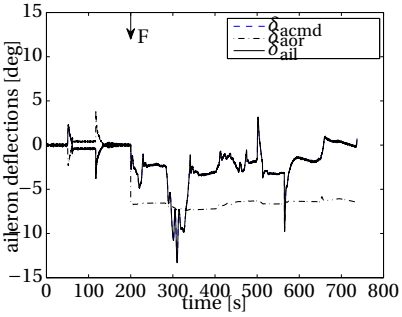


Figure 7.26: Commanded and actual aileron deflections, engine separation.

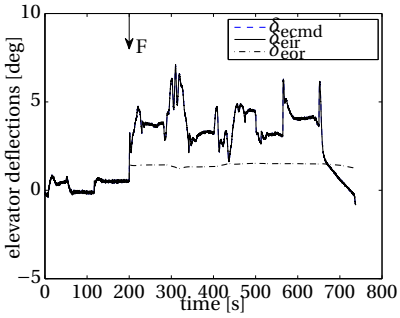


Figure 7.27: Commanded and actual elevator deflections, engine separation.

Under the engine separation failure, engine ♯1 and ♯2 remain functional. Their changing history are shown in Figure 7.20. The collective thrust P_c, i.e., mean of EPRs, is regulated to keep the true airspeed between 130 m/s and 145 m/s, see Figure 7.22. V_{TAS} is decreased after the 650$^{\text{th}}$ second, because we want the simulated flight be more similar to the real landing case of the Boeing 747-200 aircraft.

The changes of body angular rates are illustrated in Figure 7.21. It can be seen that p, q and r stay at zero during level-straight flight even though the engine separation fault occurs. In addition, the body angular rate changes caused by the fault (around the 200$^{\text{th}}$ second) are relatively small when compared to those demanded by the roll command between 50s and 100s.

The tracking performance of the joint SBB angular controller is depicted in Figs. 7.23-7.25. Zero-error tracking performance are guaranteed in the ϕ and θ channels, and β is kept at zero except a short period, i.e., from 285s to 330s. During this short period, β is non-zero due to the fact that the remaining operational authority of the rudder is quite limited since a large part of its working range has been occupied by generating the yawing moment, which is aimed at counteracting the yawing moment induced by the right engine separation. It should be noted that the aircraft needs a higher level of thrust when it is climbing, i.e., from 285s to 330s, in order to make the airspeed maintained.

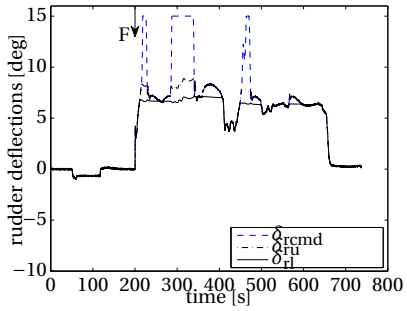


Figure 7.28: Commanded and rudder deflections, engine separation.

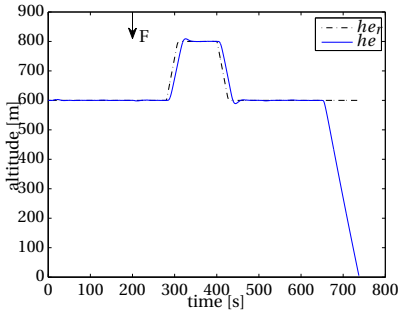


Figure 7.29: Altitude, engine separation.

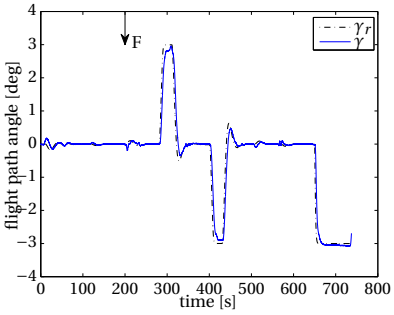


Figure 7.30: Flight path angle, engine separation.

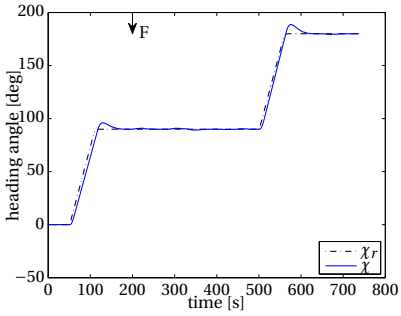
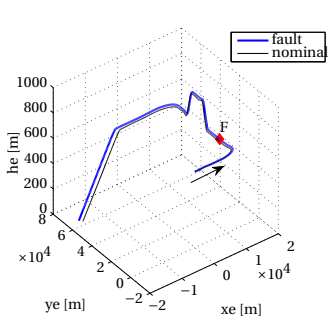
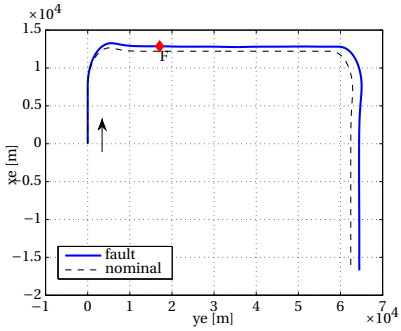


Figure 7.31: Heading angle, engine separation.



(a) Three-dimensional trajectory.



(b) Ground track.

Figure 7.32: Trajectory comparison results between the nominal case and engine separation case.

However, the increase in thrust will lead to a large yawing moment which needs to be counteracted by the rudders. Compared to the simulation results presented in [139], i.e., the dashed curve in Figure 7.25 is the same as that in [139], the joint method leads to

Table 7.2: Joint SBB attitude controller parameters, $\epsilon = 0.35$, rudder runaway

Channel	proportion	integration
Angular control	[0.25, 0.2, 0.1]	[0.1, 0.1, 0]
Body rate control	[1, 1, 1]	[0, 0, 0]

a much smaller β than that led by the hybrid SBB method during most of the time, i.e., except the period from 280s to 330s. In addition, the magnitude of the sideslip angles led by both methods are comparable under the engine separation case in the period from 280s to 330s.

The changing history of the commanded and actual deflections of the control surfaces are depicted by Figs. 7.26-7.28. The control surface deflections of ailerons, elevators and rudders are plotted separately. It can be observed from Figs. 7.26-7.27 that δ_{eor} and δ_{aor} are not active/responding under this failure. Actually, δ_{eil} is also not active/responding under this failure. In Figure 7.28, a large difference between the commanded and the actual rudder deflections shows up after the failure is triggered. It indicates that the rudders are working in a saturated state.

Figures. 7.29-7.31 provide the records of the tracking performance in the flight path control level. The closed-loop aircraft is able to track the reference commands, i.e., χ_{r}, γ_{r} and he_{r}, with zero error. It should be mentioned that the flight path controller is acting as an inner controller before the 650$^{\text{th}}$ second, which makes the actual γ not necessary to follow its reference command γ_r with zero tracking error in this stage. According to Figure 7.22 and Figure 7.29, the aircraft is climbing from 600m to 800m within 50s while sacrificing 7.5 m/s of airspeed.

The three-dimensional trajectory of the damaged aircraft, under the right engine separation case, is compared with that of the nominal aircraft, see Figure 7.32(a). The second curve marked with 'nominal' is the same as that in Figure 7.19. To increase the clarity, the ground tracks of the nominal, i.e., fault-free, aircraft and the post-failure aircraft are compared in Figure 7.32(b). As can be seen from Figure 7.32, the trajectory difference of the aircraft between the nominal case and the engine separation case is negligible. Specifically, the aircraft under the engine separation fault can fly a right turn with the same turning rate as that of the nominal aircraft, see the second turn in Figure 7.32(b).

In the second simulation experiment, the joint SBB attitude controller using the control structure shown in Figure 7.1(a) is validated using rudder runaway fault scenario. As shown in Figure 7.1(c), differential thrust is introduced to counteract the yawing moment induced by the stuck rudders. The controller parameters are selected as shown in Table 7.2.

During the simulation, the rudder runaway fault scenario was triggered at the 200$^{\text{th}}$ second. The validation results of the joint SBB angular controller are plotted in Figs. 7.33-7.46.

The changes of the total EPRs ($\mathbf{P}_{\text{t}}$) are shown in Figure 7.33, and all of them reach saturation limits immediately after the failure occurs. As shown in Figure 7.35, the true airspeed is maintained between 130 m/s and 163 m/s by controlling the collective thrust P_c. The differential thrust $\mathbf{P}_d$ is responsible for producing yawing moment needed by

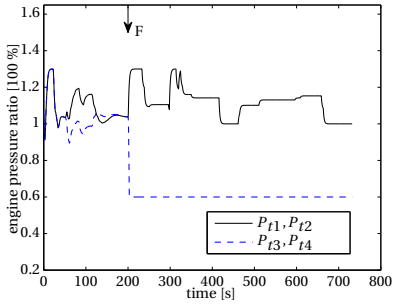


Figure 7.33: Engine Pressure Ratios, rudder runaway.

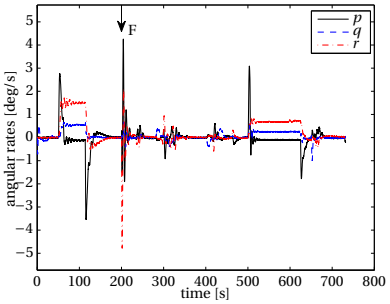


Figure 7.34: Angular rates, rudder runaway.

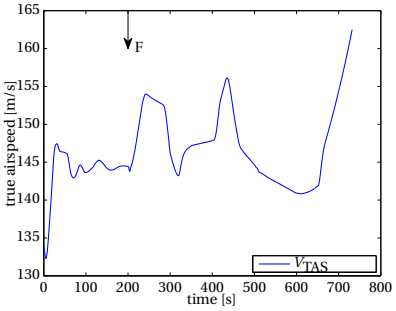


Figure 7.35: True airspeed, rudder runaway.

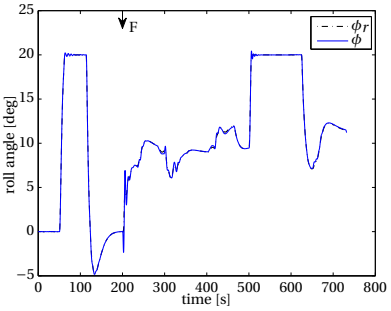


Figure 7.36: Roll angle, rudder runaway.

the flight control task. To ensure that the true airspeed is below 163 m/s after the rudder failure occurs to the aircraft, a dynamic upper limit of $\mathbf{P}_t$ is employed with its highest value equal to 1.3.

As shown in Figure 7.34, the rudder runaway failure occurring at the 200th second produces great influences on p, q and r. Both the roll rate and the pitch rate have demonstrated a spike around the 200th second. This is caused by the over-reacting of the control effectors, i.e., ailerons and differential thrust. At the beginning, the yawing moment induced by the stuck rudder drives the yawing rate r to -0.08 rad/s. Then, the yawing moment produced by the differential thrust as well as that contributed by the instantaneous nonzero β drives r to 0.037 rad/s within an extremely short time period.

The angular command tracking performance of the joint SBB controller is illustrated in Figs. 7.36-7.38. As shown in Figs. 7.36 and 7.37 respectively, the tracking errors of ϕ and θ nearly equal to zero. In Figure 7.25, the dashed curve is the same as that in [139]. As shown in Figure 7.25, β is kept between 4 deg and 8.6 deg. This is ascribed to the fact that the rudder runaway failure is so severe that it uses up all of the control authority of the differential thrust. As a consequence, β cannot be further reduced using the main control effector (the differential thrust). However, the nonzero tracking errors of β is still comparable to or even smaller than those presented in [87] and [139] under the same

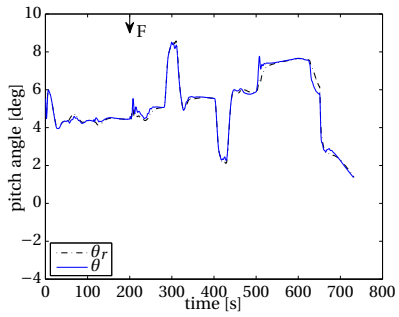


Figure 7.37: Pitch angle, rudder runaway.

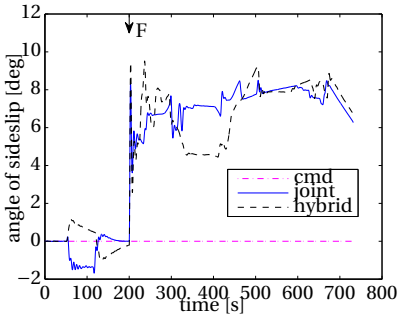


Figure 7.38: Angle of sideslip, rudder runaway.

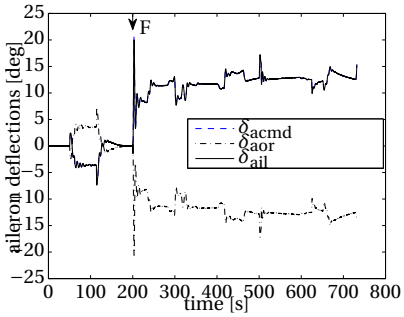


Figure 7.39: Commanded and actual aileron deflections, rudder runaway.

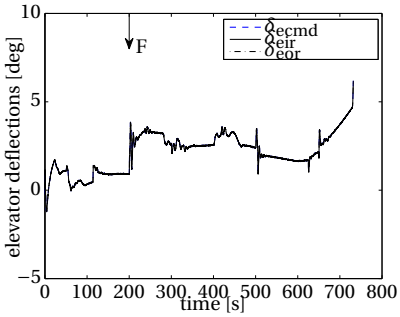


Figure 7.40: Commanded and actual elevator deflections, rudder runaway.

fault scenario.

In terms of the sideslip angle control, the joint SBB approach has an equal or a slightly better performance than the hybrid SBB approach from [139]. This is because the joint SBB angular controller leads to sideslip angles with a slower changing rate than those led by the hybrid SBB angular controller, see. Figure 7.38. That is, the non-zero sideslip angle stays around a steady position. In addition, the maximum non-zero sideslip angles produced by both methods are nearly the same. It should be kept in mind that the aircraft inevitably has to fly with a non-zero sideslip angle under the rudder runaway case due to the reduced control authority. In that sense, although Figure 7.38 shows the hybrid SBB controller produces smaller sideslip angles than those produced by the joint SBB controller from the 340^{th} second to the 410^{th} second, it still does not mean the hybrid SBB control approach has a better performance than the joint SBB control approach.

Figures. 7.39 and 7.40 shows the changes of the commanded and actual (limited by the actuator dynamics and failures) control surface deflections in the roll and pitch channel, respectively. In Figure 7.41, the changes of the actual rudder deflections are plotted. It should be noted that the deflection angles of rudders vary although they are stuck to the left limit under the rudder runaway failure. The reason of doing so can be

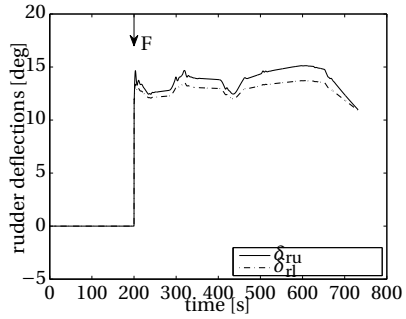


Figure 7.41: Actual rudder deflections, rudder runaway.

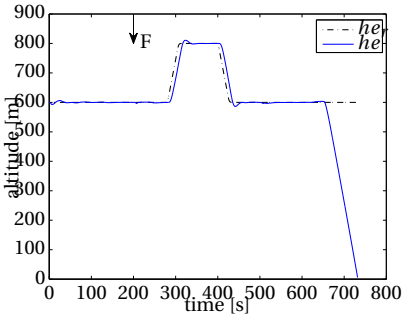


Figure 7.42: Altitude, rudder runaway.

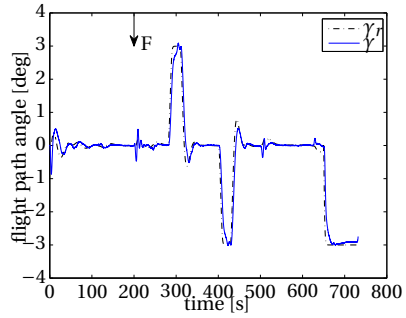


Figure 7.43: Flight path angle, rudder runaway.

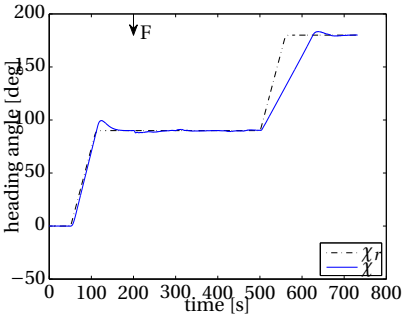


Figure 7.44: Heading angle, rudder runaway.

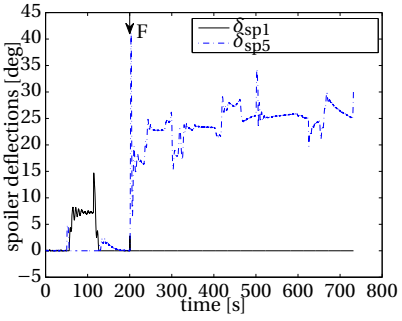
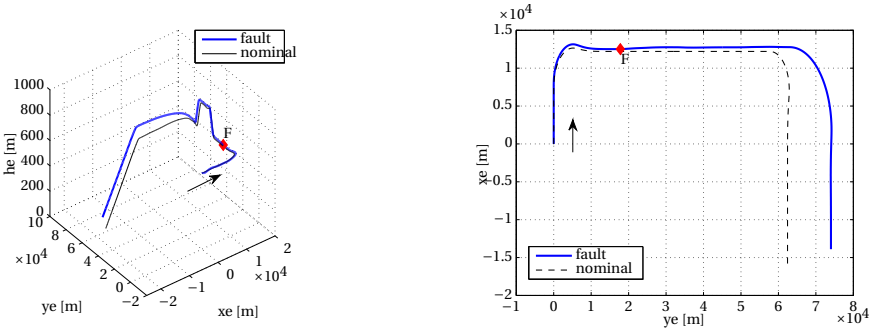


Figure 7.45: Spoiler deflections, rudder runaway.

found in [87] and [139].

As can be seen from Figs. 7.42-7.44, χ, γ and he are closely tracking their own reference command, respectively. That is, zero tracking errors are ensured in the steady state. It should be mentioned that the aircraft is under altitude control mode during the first $650s$, and it is under the flight path angle control mode after the 650^{th} s. Since the



(a) Three-dimensional trajectory.

(b) Ground track.

Figure 7.46: Trajectory comparison results between the nominal case and rudder runaway case.

flight path angle controller is an inner controller during the first 650s, the tracking errors of γ are allowed to be nonzero during a few transient periods, see Figure 7.43.

The changing history of the spoiler deflections is depicted by Figure 7.45. The spoilers are assisting the ailerons in producing roll moment. Finally, the three-dimensional trajectory of the post-failure aircraft is compared with that of the fault-free aircraft in Figure 7.46(a). Similar to Figure 7.32(b), Figure 7.46(b) provides the ground tracks correspondingly. Again, the second curve marked with 'nominal' is as that in Figure 7.19. It can be observed that the post-failure aircraft needs a wider turn than the fault-free aircraft, see the second right turn from Figure 7.46(b). This is partially due to the fact that the sideslip angle of the post-failure aircraft is nonzero. It should be noted that the bank angles of the post-failure aircraft and the nominal aircraft are not equal though the roll angles are equal (20 deg). This difference is caused by the nonzero β under the rudder runaway fault scenario.

7.6. CONCLUSIONS

This chapter has extended the singular perturbation theory based sensor based backstepping (SBB) method presented in [46] to design a double-loop angular controller with a focus on handling the sudden structural changes associated with the benchmark failures. The double-loop SBB attitude controller is designed using the backstepping technique, which guarantees the stability of the quasi-cascade subsystem, i.e., the body angular rate dynamics and the angular motion dynamics. Another advantage of the proposed method is that it does not require real-time full model information of an aircraft.

The double-loop SBB angular controller is combined with an outer-loop flight path controller designed using regular proportional-integral-derivative (PID) control laws. The overall autopilot flight controller has four levels of control loops, and was evaluated using a Boeing 747-200 aircraft under the nominal case and two benchmark fault scenarios, i.e., rudder runaway and engine separation, developed by the GARTEUR FM-AG 16 group. In dealing with the rudder runaway fault scenario, the differential thrust

is introduced to compensate the unexpected yawing moment induced by the failed rudders.

The numerical simulation results show that the double-loop joint SBB angular controller can lead to zero tracking errors as long as the given angular reference commands are within the safe flight envelope. Compared with the hybrid SBB angular controller presented in [139], the new joint SBB method leads to better zero-hold performance in controlling sideslip angle when the aircraft is flying in the nominal condition or under the engine separation scenario. Under the rudder runaway fault scenario, the new method presented in this chapter leads to equivalent control performance of sideslip to that of the hybrid SBB control method mentioned above.

8

TENSOR-PRODUCT SIMPLEX SPLINES FOR AERODYNAMIC MODEL STRUCTURE SELECTION AND ADAPTIVE CONTROL

Standard simplex B-spline based model identification methods were studied with a focus on enhancing the computational efficiency of the parameter estimation process in Chapters 3 and 4. This chapter is aimed at providing more flexible options for choosing model structures when using multivariate simplex B-splines. A tensor-product simplex B-spline model structure was extended from a single dimension case into a multidimensional case. More *a priori* knowledge of the the model can be taken into account when defining spline model structures. Two double-loop attitude controller design methods, namely a hybrid controller and a joint controller were presented in Chapters 6 and 7. Central to each controller is an acceleration measurement-based incremental nonlinear control (AMINC) law. This control approach contains a time-scale parameter which is related to a control effectiveness matrix, so taking into consideration the estimation of the control effectiveness matrix, the tensor-product simplex B-spline method was used for this purpose.

Recently, recursive function approximation approaches using multivariate simplex B-splines have been investigated for the aim of providing an accurate global aerodynamic model for use in adaptive flight control systems. To enable a real-time implementation, the efficiency of this model identification approach still needs to be enhanced without sacrificing modeling accuracy. In this chapter, a new mathematical procedure for constructing spline models is presented, which provides more options for selecting an efficient model structure. For this, a new multivariate spline approach indicated as the tensor-product simplex B-splines is applied. Using this new approach, more a priori *knowledge*

can be taken into account when defining model structures. The new tensor-product simplex B-splines can lead to a lower computational load compared to standard simplex B-splines because fewer global B-coefficients are required for comparable approximation power. Most importantly, the new spline approach maintains the natural feature of the multivariate simplex B-splines in the sense that the global B-form regression vector is normalized, each basis polynomial is guaranteed to be well-conditioned numerically, and differentiability is maintained along each input dimension. An F-16 nonlinear model is used to generate data to validate the new approach. Simulation results show that the new approach can achieve a higher level of approximation accuracy using less parameters when modeling the aerodynamic moment coefficients, and can provide proper and bounded estimations of the control effectiveness matrix, which is comprised of the directional derivatives of the output of the spline functions.

8.1. INTRODUCTION

A powerful model identification method is required when designing a model-based adaptive flight control system. Among all of the model identification methods found in the literature, one promising option is the multivariate simplex B-splines (MVSB) method, which is a parametric functional approximator [10, 34, 37, 73]. In [34], a batch method using multivariate simplex B-splines is proposed. In 2011, de Visser et al. [37] proposed a recursive simplex B-spline method indicated as the equality constrained recursive least squares (ECRLS) MVSB. As an extension to the regular smoothness constraints on the B-coefficients, de Visser et al. [37] also take into account the differential constraints in propagating the spline model to avoid the divergence around the boundary of the spline model during extrapolation. Sun et al. [138] present a substitution strategy to enhance the computational efficiency of the recursive simplex B-spline method presented in [37].

Simplex B-splines have the following benefits. Firstly, they give a higher approximation power than regular polynomials as the approximation power is dominated by both the density of simplices and the polynomial order within each simplex [34]. Secondly, a continuous global model is provided by enforcing continuity between spline subdomains [36, 73]. Finally, the output of the aerodynamic model is bounded once the B-coefficients of the spline model is bounded. This property is attributed to the partition-of-unity property in that the B-form regression vector is normalized with each of its elements non-negative. These three benefits enable the online global aerodynamic model identified using the simplex B-splines to have a high approximation power and a high level of reliability. Since enhancing the accuracy of an aerodynamic model usually helps to enhance the power of a model-based adaptive flight controller, an aircraft equipped with a simplex B-spline model-based controller has the potential to achieve a high level of flight performance during maneuvers [155, 157].

When sufficient *a priori* knowledge is available on the physics of the system, using standard simplex B-splines is not optimal, because they will introduce large numbers of basis polynomials in the B-form regression vector which have no clear physical meaning. This is essentially due to the inherent nature of the standard simplex B-splines that each dimension of the inputs is treated equally when constructing the global B-form basis vector. To optimize the model structure and reduce the computational load for

the purpose of real-time model identification, the total number of both the simplices and the global B-coefficients can be reduced further by choosing another structure for the simplex B-splines. Typical examples where the model structure of the simplex B-splines can be optimized include: 1) a certain dimension of the function input is affine to the outputs or is less coupled with other inputs according to *a priori* knowledge. 2) the highest polynomial order of a certain input dimension is known and much lower than that of the other input dimensions according to *a priori* knowledge. One such example is the aerodynamic model of F-16 aircraft. For an F-16, the aerodynamic force and moment coefficients are affine to the control surface deflections, and the highest polynomial orders required for the control inputs are relatively low [114].

An accurate and fast control allocation (CA) solver plays an important role in model-based adaptive flight control [39]. The CA problem associated with an adaptive nonlinear dynamic inversion (NDI) controller which uses a polynomial structure has been thoroughly investigated, see [39, 87, 163]. However, the literature on investigating how the simplex B-spline based aerodynamic models can be incorporated into an adaptive nonlinear flight control system is still limited. In Tol et al. [157], three CA solvers for an adaptive NDI framework are presented: a linear CA solver, a successive linear CA solver and a nonlinear CA approach. All of them are developed based on a global re-parameterization of the spline function, i.e., the spline function is transformed from the barycentric coordinates to the global Cartesian coordinates. Furthermore, because the parametric aerodynamic model identified using the standard simplex B-splines [37] is non-affine in the control inputs, the CA unit requires either a local linearization technique or a parameterization technique. For all three CA solvers, the gradient of the spline function needs to be calculated once, or a number of times, at each CA step.

Standard simplex B-splines have a shortcoming during control allocation for the purpose of flight control. This is that, in the successive linear CA solver and the nonlinear CA solver [157], a large number of simplices neighboring to the current operating point are involved, and a local optimum needs to be determined for each simplex. This is due to the fact that the control inputs are highly coupled with the aircraft states in the standard simplex spline model structure. The control inputs are treated equally, compared to the aircraft states, and are equally involved in partitioning the entire function input space. Once the control input changes, the evaluation point may enter another subdomain even if the control input stays in the same partitioning in that single dimension.

To solve the shortcoming of using standard simplex B-splines mentioned above, a new compound structure is developed for the multivariate simplex B-splines in this chapter. The new approach is referred to as the multivariate tensor-product (TP) simplex B-splines in the remainder of this chapter because the tensor product operator is applied when deriving the global B-form basis vector. The concept of the TP simplex B-splines is presented in Govindarajan et al. [54], in this chapter we extend the new spline approach to include any number of spatial dimension tensor products. The properties and application issues of the TP simplex B-splines will be discussed. Specifically, we investigate the application of the new method to the identification of the aerodynamic moment coefficients.

The new TP simplex spline allows the decomposition of the global model domain

into smaller subsets of lower dimension. Each of the lower dimensional subsets can support a standard simplex B-spline function. The global TP simplex spline function is then formed by taking the tensor product of the lower dimensional splines. For example, based on expert knowledge of the aerodynamics of an aircraft, the user may decide to model the aerodynamic moment coefficients using bivariate simplex splines, while the influence of the control surfaces is modeled using only a univariate spline function. The complete TP simplex spline basis is then formed by taking the tensor product between the bivariate, and univariate splines thereby forming a 3-dimensional basis with limited cross-couplings. To apply this approach the function output of a system is assumed to be affine in a certain dimension of inputs or the highest polynomial orders required for certain dimensions are assumed to be lower than those of other dimensions. In these dimensions, a lower dimensional simplex B-spline basis is defined. Due to the application of multivariate TP simplex B-splines, the overall multidimensional control input space is divided into multiple sub domains, which are not strict simplices.

Tensor-product simplex B-splines have a number of advantages compared to standard simplex B-splines. Firstly, more *a priori* knowledge of the system under consideration can be taken into account in a more proper way by the TP simplex B-splines since this method allows for more flexible model structure selection. Secondly, the model structure and the polynomial terms in the basis regression vector have a more transparent physical interpretation because some high-order basis terms, which may be considered un-physical for a certain inputs, can be discarded by tensor-product simplex B-splines. Thirdly, the computational load of the simplex B-spline identification method is reduced, given the same model dimension and the highest polynomial order of each dimension, due to the reduction of the length of the global basis regression vector. Finally, tensor-product simplex B-splines require a lower number of training data attributed to the reduced length of the global basis regression vector and the global B-coefficient vector.

The objective of this chapter is to present a new generalized tensor-product structure for multivariate simplex B-splines, which provides *more options* for model structure selection when using the splines. The crux is that this new spline gives the user the freedom to use expert knowledge of the system to influence the model structure selection; something that is infeasible using standard simplex B-splines. The control effectiveness matrix for an F-16 aircraft can be derived conveniently from a TP simplex spline model with higher or equivalent accuracy compared to standard simplex B-splines. Additionally, the optimization algorithm in the control allocation block can be simplified because the TP simplex B-spline structure produces less sub domains in the dimensions of the control inputs and requires fewer B-coefficients than the standard simplex B-spline model structure. These features allow the tensor-product simplex B-splines to be implemented in real-time, i.e., on board, with lower computational load requirements compared to an equivalent standard simplex spline, see [37].

The structure of this chapter is as follows. The problem statement will be given in Section 8.2. Preliminaries on multivariate simplex B-splines will be given in Section 8.3. The new tensor-product simplex B-splines and their properties are presented in Section 8.4. In Section 8.5, the model structure selection for an F-16 aircraft is discussed, and tensor-product (TP) simplex B-splines are compared to standard simplex

B-splines in terms of approximation power and computational load. In Section 8.6, a data set obtained from a simulated flight trajectory is used to validate the new model structure further, and the calculation of the control derivatives is validated. The TP simplex B-splines are used to estimate a control effectiveness matrix for an incremental backstepping controller in Section 8.7. This chapter is concluded in Section 8.8.

8.2. PROBLEM STATEMENT

The main focus of this chapter is online identification of an aerodynamic model for an F-16 aircraft using the multivariate tensor-product simplex B-splines. This model identification routine focuses on the second step of the two-step method, and turbulence on the aerodynamic model of an F-16 aircraft is not taken into account in this chapter. In addition, the calculation of the control effectiveness matrix is discussed in detail and the optimization based control allocation process is introduced briefly. Similar to Tol et al. [155], the aerodynamic model identification block is aimed at providing parameter adaptation in real time for an adaptive nonlinear dynamic inversion (NDI) body angular/angular rate control system. The spline model-based NDI control system is briefly introduced in this section for the purpose of making the tensor-product simplex B-spline model structure easier to understand. But it should be kept in mind that investigation on spline model-based control system is not the focus of this chapter and thus not included in the remaining sections; for this we refer to the literature [157].

The overall framework of the adaptive nonlinear dynamic inversion (NDI) based body angular rate flight control system is shown in Figure 8.1. In this figure, 'CA' is the acronym for 'control allocation'; 'IEKF' stands for 'iterative extended Kalman filter'; 'AMI' represents 'aerodynamic model identification'; 'LC' denotes 'linear controller'. The function $a(x)$ represents the state, i.e., x, contributed part of the spline model, the function $b(x)$ represents the part of the spline model which is contributed by the control inputs.

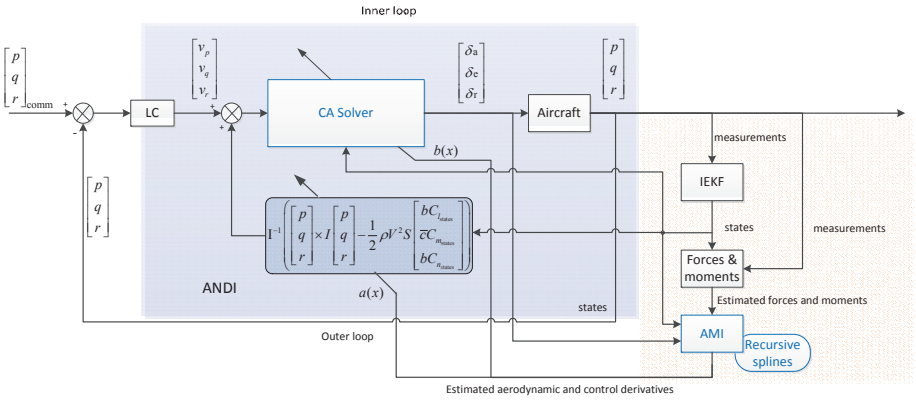


Figure 8.1: Rate control structure using adaptive nonlinear dynamic inversion (ANDI) with the recursive spline method located in the aerodynamic model identification (AMI) block.

In this chapter, we mainly focus on the aerodynamic model identification block, i.e., the block named 'AMI' in Figure 8.1, and discuss the 'CA Solver' block briefly. The aerodynamic model identification method using multivariate simplex B-splines designed in the 'AMI' block should be able to provide an accurate real-time model for the purpose of adapting the controller parameters. The developed identification approach should allow the control allocation algorithm, i.e., in the 'CA' solver block, to be as simple and precise as possible. To accomplish the control allocation, the calculation of the control effectiveness matrix from the spline model will be discussed in detail.

8.2.1. AERODYNAMIC EQUATIONS AND PRELIMINARIES ON AIRCRAFT MODEL IDENTIFICATION

For a fixed wing aircraft such as F-16, the following rotational dynamic equations hold:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = -\left\{ \mathbf{I}^{-1} \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right) - \frac{1}{2} \rho V^2 S \cdot \mathbf{I}^{-1} \begin{bmatrix} bC_{l_{\text{states}}} \\ \bar{c} C_{m_{\text{states}}} \\ bC_{n_{\text{states}}} \end{bmatrix} \right\} + \frac{1}{2} \rho V^2 S \cdot \mathbf{I}^{-1} \mathbf{M}_{\text{CA}} \cdot \mathbf{u} \tag{8.1}$$

where $\mathbf{u} = [\delta_a, \delta_e, \delta_r]^\top$, and the control allocation matrix $\mathbf{M}_{\text{CA}}$ is defined as follows:

$$\mathbf{M}_{\text{CA}} = \begin{bmatrix} b & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & b \end{bmatrix} \mathbf{M}_E = \begin{bmatrix} b & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & b \end{bmatrix} \begin{bmatrix} C_{\text{L}_{\delta_a}} & C_{\text{L}_{\delta_e}} & C_{\text{L}_{\delta_r}} \\ C_{\text{M}_{\delta_a}} & C_{\text{M}_{\delta_e}} & C_{\text{M}_{\delta_r}} \\ C_{\text{N}_{\delta_a}} & C_{\text{N}_{\delta_e}} & C_{\text{N}_{\delta_r}} \end{bmatrix} \tag{8.2}$$

As a preparation for the aerodynamic model identification process, the non-dimensional moment coefficients can be reconstructed using the measured states as follows:

$$C_{\text{l}} = \frac{L}{\frac{1}{2}\rho V^2 S b} = \frac{\dot{p} I_{xx} + qr\left(I_{zz} - I_{yy}\right) - \left(pq + \dot{r}\right) I_{xz}}{\frac{1}{2}\rho V^2 S b} \tag{8.3}$$

$$C_{\text{m}} = \frac{M}{\frac{1}{2}\rho V^2 S \bar{c}} = \frac{\dot{q} I_{yy} + rp\left(I_{xx} - I_{zz}\right) - \left(p^2 - r^2\right) I_{xz}}{\frac{1}{2}\rho V^2 S \bar{c}} \tag{8.4}$$

$$C_{\text{n}} = \frac{N}{\frac{1}{2}\rho V^2 S b} = \frac{\dot{r} I_{zz} + pq\left(I_{yy} - I_{xx}\right) - \left(qr - \dot{p}\right) I_{xz}}{\frac{1}{2}\rho V^2 S b} \tag{8.5}$$

In aerodynamic model identification for an F-16, 8 input dimensions are possibly required for modeling C_{M}:

$$[\alpha, \beta, \frac{q\bar{c}}{V}, \delta_e, \delta_{\text{lef}}, \delta_{\text{T}}, \frac{pb}{2V}, \frac{rb}{2V}] \tag{8.6}$$

Similarly, there are 8 possible input arguments for C_{L} and C_{N}:

$$[\alpha, \beta, \frac{pb}{2V}, \frac{rb}{2V}, \delta_a, \delta_r, \delta_{\text{lef}}, \frac{q\bar{c}}{V}] \tag{8.7}$$

8.2.2. CONTROL ALLOCATION RELATED TO THE SIMPLEX B-SPLINES

In the standard simplex B-splines, each dimension of the inputs will be treated equally and each input is involved in the triangulation, i.e., partitioning the whole input space into multiple sub domains. A 2-D example is shown in Figure 8.4 where 'x' is a scalar state and 'u' is a single input. The solid point on the dashed vertical line denotes the current working point with the current state $x_0 = 0.5$. Then, we start to seek an optimal input u along the dashed vertical line. The problem now is that the optimum could be located in all the four simplices on the left-hand side [157]. When using the successive linear CA solver or the nonlinear CA solver from [157], all these four neighboring piece-wise models need to be parameterized and a local optimum needs to be calculated for each per-simplex local model. If the state x and input u are of higher dimensions, the number of the piece-wise models involved in the CA process will grow exponentially. This will make the global optimization process in the control allocation solver very complex. To this end, a new compound structure is introduced for the simplex B-splines in this chapter, the new spline model structure is referred to as tensor product (TP) simplex B-splines in the remainder of this chapter. Compared to the standard simplex spline model structure, less local piece-wise models are involved in the optimization at one sample step when using the TP simplex B-spline model structure. This is ascribed to the fact that the control inputs are treated differently from other system states, and the control inputs are less involved in partitioning the entire function input space when using the TP simplex B-spline model structure.

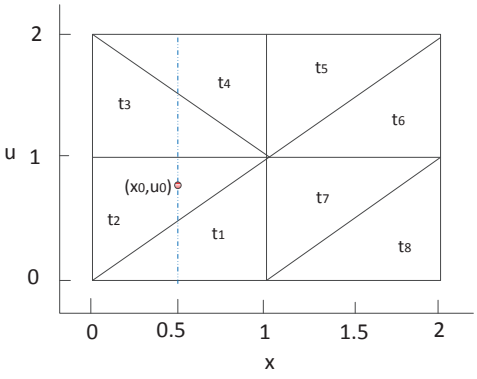


Figure 8.2: Control allocation optimization using the bi-variate simplex B-splines, single state x, single input u [155].

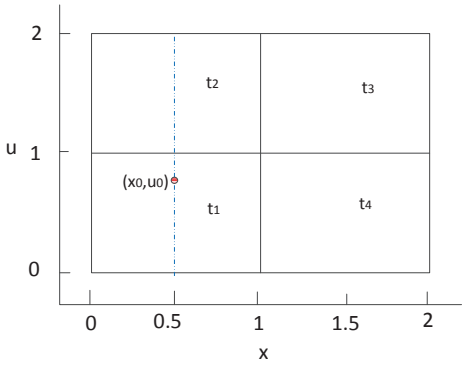


Figure 8.3: Control allocation optimization using the tensor-product simplex B-splines, single state x, single input u, .

As shown in Figure 8.3, the TP simplex B-splines lead to four rectangular subdomains. Recall that standard simplex B-splines in this case lead to eight simplices, see Figure 8.4. A 3-D example is shown in Figure 8.4. In the 3-D case, when using standard simplex B-splines, each subdomain has a shape of tetrahedron. In contrast, when using TP simplex B-splines, each subdomain is a triangular prism.

Existing literature in which the control allocation problem related to the multivariate simplex B-spline aerodynamic model is discussed is limited to [155, 157]. Two types of optimization based control allocation approaches are presented in [157]. The first one

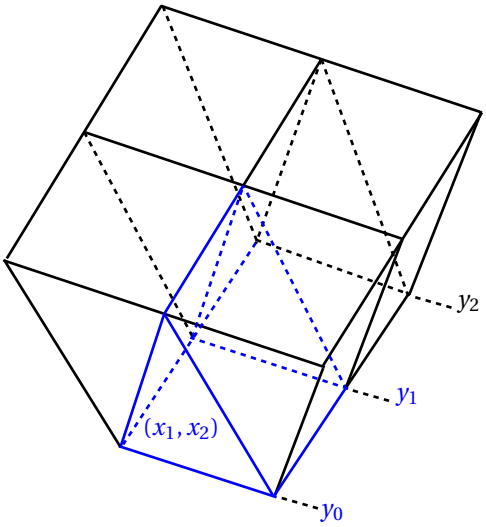


Figure 8.4: Piece-wise model partitioning in tensor-product simplex B-splines, 3-D demonstration, [54].

is indicated as the local linearization based control allocation approach, because the global spline function is locally linearized around a simplex by calculating the directional derivatives in terms of the inputs. The second is referred to as the parameterization based control allocation in the remainder of this chapter, because it parameterizes the spline function on a specific simplex, i.e., current working point, in terms of global coordinates, i.e., Cartesian coordinates. Then, the cost function of the optimization problem is calculated from the parameterized spline model. In the first approach, it is assumed that the linearized model stands for the whole input space with the arguments defined in the Cartesian coordinates. In the second approach, only a locally-valid spline model is derived by parameterizing the global spline model on a specific simplex, which covers the current work point or is its neighbors.

To accomplish the second control allocation approach mentioned above, a B-form basis vector in the global coordinates, i.e., Cartesian coordinates, needs to be calculated according to [33][p.148-158]. If the second control allocation approach is applied to standard multivariate simplex B-splines, it usually yields multiple local optimum solutions among the neighboring simplices of the current working point.

The TP simplex B-splines with a compound structure are well suited to approximate aerodynamic force and moment coefficients. One reason is that the aerodynamic force and moment coefficients are affine to certain control inputs in many cases, e.g., the roll moment of an F-16 aircraft is affine to the aileron deflections according to the expert knowledge. Secondly, these compound splines allow for treating aircraft states and control inputs in a different manner, which can simplify the control allocation approach. The standard nonlinear model of the F-16 fighter aircraft taken from [114] is used to validate the TP simplex B-splines approach presented in this chapter.

8.3. PRELIMINARIES ON MULTIVARIATE SIMPLEX B-SPLINES

For the purpose of completeness, the preliminaries of the regular simplex B-splines will be introduced briefly in this section. For more details on multivariate simplex B-splines, the reader is referred to [34, 37, 73].

8.3.1. TRIANGULATIONS, BARYCENTRIC COORDINATES AND PIECE-WISE BASIS FUNCTION

In essence, a simplex B-spline function is a piece-wise polynomial function with continuity constraints enforced between different pieces. The approximation power of an multivariate simplex spline function is partly determined by the triangulation. A triangulation $\mathscr{T}$ is a special partitioning of a domain into a set of J non-overlapping simplices:

$$\mathscr{T} := \bigcup_{i=1}^{J} t_i,\ t_i \cap t_j \in \{\varnothing, \tilde{t}\}, \forall\, i \neq j,\ t_i, t_j \in \mathscr{T} \tag{8.8}$$

with the edge simplex $\tilde{t}$ a k-simplex with $0 \leq k \leq n-1$. The n-simplex denoted by t_i or t_j is formed by the convex hull of its $n+1$ non-degenerate vertices $(v_0, v_1, ..., v_n) \subset \mathbb{R}^n$. High quality triangulations can be obtained using constrained Delaunay triangulation (CDT) methods, such as the 2-dimensional CDT method presented in Shewchuk [128].

For the purpose of making later citation easier, a single n-simplex is written as t after discarding the subscript. The normalized barycentric coordinates of an evaluation point $\mathbf{x} \in \mathbb{R}^n$ with respect to simplex t are defined as:

$$b(\mathbf{x}) \quad := \quad (b_0, b_1, ..., b_n) \in \mathbb{R}^{n+1},\ \ \mathbf{x} \in \mathbb{R}^n \tag{8.9}$$

which follows from the following implicit relation:

$$\mathbf{x} = \sum_{i=0}^{n} b_i v_i,\ \sum_{i=0}^{n} b_i = 1 \tag{8.10}$$

According to [73] and [34], the Bernstein basis polynomial $B_\kappa^d(b(\mathbf{x}))$ of degree d in terms of the barycentric coordinates $b(\mathbf{x}) = (b_0, b_1, ..., b_n)$ from Eq. 8.10 is defined as:

$$B_\kappa^d(b(\mathbf{x})) := \begin{cases} \frac{d!}{\kappa_0!\kappa_1!\cdots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \cdots b_n^{\kappa_n} & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases} \tag{8.11}$$

where $\kappa = (\kappa_0, \kappa_1, ..., \kappa_n) \in N^{n+1}$ is a *multi-index* with the following properties: $\kappa! = \kappa_0!\kappa_1!...\kappa_n!$ and $|\kappa| = \kappa_0 + \kappa_1 + ... + \kappa_n$. In Eq. 8.11, we use the notation $b^\kappa = b_0^{\kappa_0} b_1^{\kappa_1} ... b_n^{\kappa_n}$. Given that $|\kappa| = d$, the total number of valid permutations of the *multi-index* κ is:

$$\hat{d} = \frac{(d+n)!}{n!d!} \tag{8.12}$$

In [30], it is proved that any polynomial $p(b)$ of degree d on a simplex t can be written as a linear combination of $\hat{d}$ basis polynomials in what is known as the B-form as follows:

$$p^t(b(\mathbf{x})) := \begin{cases} \sum_{|\kappa|=d} c_\kappa^t B_\kappa^d(b(\mathbf{x})) & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases} \tag{8.13}$$

with c_κ^t the B-coefficients which uniquely determines $p^t(b(\mathbf{x}))$, where the superscript 't' indicates that p is defined on the simplex 't'. The total number of basis function terms is equal to $\hat{d}$, which is the total number of valid permutations of κ.

8.3.2. VECTOR FORMULATIONS OF THE B-FORM

As introduced in [37], the vector formulation, according to Eq. 8.13, for a piece-wise B-form polynomial $p(b(\mathbf{x}))$ in barycentric $\mathbb{R}^{n+1}$ has the following expression:

$$p^t(\mathbf{x}) := \begin{cases} \mathbf{B}_t^d(b(\mathbf{x})) \cdot \mathbf{c}^t & ,\mathbf{x} \in t \\ 0 & ,\mathbf{x} \notin t \end{cases}, \tag{8.14}$$

with $b(\mathbf{x})$ the barycentric coordinates of the Cartesian $\mathbf{x}$. The row vector $\mathbf{B}_t^d(b(\mathbf{x}))$ in Eq. 8.14 is constructed from each individual basis polynomials which are sorted lexicographically [37] (see. Eq. 8.11).

The simplex B-spline function $s_d^m(b(\mathbf{x}))$ of degree d and continuity order m, defined on a triangulation consisting of J simplices, is defined as follows:

$$s_d^m(\mathbf{x}) := \mathbf{B}_g^d(b(\mathbf{x})) \cdot \mathbf{c} \ \in \mathbb{R}, \tag{8.15}$$

with $\mathbf{B}_g^d(b(\mathbf{x}))$ the global vector of basis polynomials which has the following full expression:

$$\mathbf{B}_g^d(b(\mathbf{x})) := [\ \mathbf{B}_{t_1}^d(b(\mathbf{x})) \quad \mathbf{B}_{t_2}^d(b(\mathbf{x})) \quad \cdots \quad \mathbf{B}_{t_J}^d(b(\mathbf{x}))\ ] \in \mathbb{R}^{1 \times J \cdot \hat{d}} \tag{8.16}$$

The subscript 'g' denotes 'global'. Note that, according to Eq. 8.14, we have $\mathbf{B}_{t_j}^d(b(\mathbf{x})) = 0$ for all evaluation locations $\mathbf{x}$ that are located outside of simplex t_j. This results in that $\mathbf{B}^d$ is a sparse row vector.

The global vector of B-coefficients $\mathbf{c}$ in Eq. 8.15 has the following formulation:

$$\mathbf{c} := \begin{bmatrix} \mathbf{c}^{t_1\top} & \mathbf{c}^{t_2\top} & \cdots & \mathbf{c}^{t_J\top} \end{bmatrix}^\top \in \mathbb{R}^{J \cdot \hat{d} \times 1} \tag{8.17}$$

with each $\mathbf{c}^{t_j}$ a per-simplex vector of lexicographically sorted B-coefficients.

8.3.3. GLOBAL CONTINUITY CONSTRAINTS

To keep the smoothness of the spline model among all sub domains, the following equality constraints should be maintained during the calculation of the global B-coefficient vector $\mathbf{c}$:

$$\mathbf{H} \cdot \mathbf{c} = 0 \tag{8.18}$$

with $\mathbf{H} \in \mathbb{R}^{(E \cdot R) \times (J \cdot \hat{d})}$ the smoothness matrix [34][10], the R is the number of continuity conditions per edge, and E is the number of edges in the specified triangulation.

8.3.4. THE DIRECTIONAL DERIVATIVE OF B-FORM POLYNOMIALS

The $m-th$ order derivative of the B-form polynomial $p(b(\mathbf{x}))$ on a single simplex t in the direction $\mathbf{u} \in \mathscr{R}^n$ is [37]:

$$\begin{aligned}\mathscr{D}_{\mathbf{u}}^{m} p^{t_j}(b(\mathbf{x})) &= \frac{d!}{(d-m)!} \sum_{|\kappa|=d-m} c_{\kappa}^{t,(m)}(a(\mathbf{u}))\, \mathbf{B}_{\kappa}^{d-m}(b(\mathbf{x})) \\ &= \frac{d!}{(d-m)!} \mathbf{B}^{d-m}(b(\mathbf{x}))\, \mathbf{P}^{d,d-m}(a(\mathbf{u})) \cdot c^{t_j}\end{aligned} \tag{8.19}$$

with $\mathbf{P}^{d,d-m}$ the de Casteljau matrix transforming the B-coefficients from degree d to $d-m$. To calculate the barycentric coordinates of the unit directional vector $\mathbf{u}$, i.e., $a(\mathbf{u})$, a point $\mathbf{v}$ located in simplex t_j is usually chosen, and the other point is determined by $\mathbf{w} = \mathbf{v} - \mathbf{u}$. Then we get $a(\mathbf{u}) = b(\mathbf{v}) - b(\mathbf{w})$. The following property holds: $\sum_{k=1}^{n+1} a_k(\mathbf{u}) = 0$.

8.4. TENSOR-PRODUCT COMPOUND STRUCTURE FOR SIMPLEX SPLINES

8.4.1. CRITERION FOR CHOOSING TENSOR-PRODUCT SIMPLEX B-SPLINES

In a standard simplex B-spline function, each argument with regard to each function dimension will be treated equally. For example, if one intends to include the term $x^2 \cdot y^2$ in a polynomial basis vector, the per-simplex polynomial order needs to be set as $d = 4$ at minimum. However, during the model identification process of a certain physical systems, it may be known from the a priori knowledge that the polynomial order of each function dimension is not required to be higher than a certain value, e.g., equal or smaller than 2. That is, when using the standard simplex B-splines, many terms in the basis polynomial vector, for example x^3, x^4 y^3 and y^4, have no contribution to the function outputs and therefore are not desirable. In fact, these terms may even degrade the physical correctness of the model by fitting measurement noise. In this case, these terms can be removed from the model using a tensor-product (TP) simplex B-spline model structure, where the B-form global basis vector maintains normalized.

To demonstrate the case for tensor-product simplex B-splines, a typical illustration example is given below.

Example 1:

Given a system which can be modeled using a simplex B-spline function $f = s_d^m(x,u)$ with x and u two arguments. Suppose that u is affine to the output f or the required order of u is lower than that of x. In addition, it is assumed known that the polynomial order of x is up to n_1, and the polynomial order of u is up to n_2. □

In this case, if we use a standard simplex B-spline model, it would yield a two dimensional simplex B-spline function: $s_{(n_1+n_2)}^m(x,u) = \mathbf{B}_g^{(n_1+n_2)}(b(x,u)) \cdot \mathbf{c}_{std}$. Instead, if we apply the tensor-product simplex B-splines to model the same system, the spline function has the following expression: $s_{(n_1+n_2)}^m(x,u) = \left[\mathbf{B}_g^{n_1}(b(x)) \otimes \mathbf{B}_g^{n_2}(b(u))\right] \cdot \mathbf{c}_{tp}$ which results in much less polynomial basis.

If we transform the simplex B-splines from the barycentric coordinates back into the Cartesian coordinates, the standard spline contains the terms containing $x^{n_1+1}, x^{n_1+2}, \ldots, x^{n_1+n_2}$ and $u^{n_2+1}, u^{n_2+2}, \ldots, u^{n_2+n_1}$, which are actually not desirable and perhaps even unphysical. The TP simplex spline only contains terms containing $x^1, x^2, \ldots, x^{n_1}$ and $u^1, u^2, \ldots, u^{n_2}$ For example, if $n_1 = n_2 = 2$ then the regression vector for TP simplex B-splines contains

terms x_1^2, x_1^2u, $x_1^2u^2$, x_1, x_1u, x_1u^2, 1, u, u^2. A regression vector for standard MVSB includes the following six extra terms: x^4, x^3u, x^3, u^3, xu^3, u^4. Therefore, each local regression vector, which belongs to a local simplex or sub domain, generated using the tensor-product simplex B-splines contains a lower number of polynomial terms than that generated using standard simplex B-splines when identifying a system described by **Example 1**.

8.4.2. GENERALIZED STRUCTURAL DEFINITION FOR TENSOR-PRODUCT SIMPLEX B-SPLINES

The model structure of TP simplex B-splines can be extended into arbitrary dimension and arbitrary combination. For example, an input vector $\mathbf{x}_n \in \mathbb{R}^n$ can be arbitrarily split as follows: $\mathbf{x}_n = \left[\mathbf{x}_1^\top, \mathbf{x}_2^\top, \cdots, \mathbf{x}_N^\top\right]^\top$ with $\mathbf{x}_i \in \mathbb{R}^{n_i}$ and $\sum_{i=1}^{N} n_i = n$. For simplicity, a general definition for the structure of the TP simplex B-splines is given below:

Structural Definition for tensor-product simplex B-splines:
Given an n-dimension function, a function approximator using tensor-product simplex B-splines could be denoted as $s_{\text{tp}}(n) = s_{\text{tp}}(n_1, n_2, \cdots, n_{\text{N}})$ with $n = n_1 + n_2 + \cdots + n_{\text{N}}$.

8.4.3. PROPERTY OF TENSOR-PRODUCT SIMPLEX B-SPLINES

Given an n_1-dimension vector $\mathbf{x}$, a multidimensional polynomial function has the following expressions:

$$p^{t_j}(\mathbf{x}) := \mathbf{B}_{t_j}^{d_x}(b(\mathbf{x})) \cdot \mathbf{c}_x^{t_j},\ \mathbf{c}_x \in \mathbb{R}^{J_x \cdot \hat{d}_x}, \mathbf{x} \in t_j \tag{8.20}$$

with $b(\mathbf{x})$ in barycentric $\mathbb{R}^{n_1+1}$.
Similarly, given an n_2-dimension vector $\mathbf{y}$, its univariate polynomial function correspondingly are as follows:

$$p^{t_i}(\mathbf{y}) := \mathbf{B}_{t_i}^{d_y}(b(\mathbf{y})) \cdot \mathbf{c}_y^{t_i},\ \mathbf{c}_y \in \mathbb{R}^{J_y \cdot \hat{d}_y}, \mathbf{y} \in t_i \tag{8.21}$$

with $b(\mathbf{y})$ in barycentric $\mathbb{R}^{n_2+1}$.

The global vector formulation of the spline basis can be derived using Eq. 4.9. Let them be written as $\mathbf{B}_g^{d_x}(b(\mathbf{x}))$ and $\mathbf{B}_g^{d_y}(b(\mathbf{y}))$ respectively. Then, the global vector of the B-form spline basis with regard to both arguments $\mathbf{x}$ and $\mathbf{y}$ are derived as follows:

$$\mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y})) = \mathbf{B}_g^{d_x}(b(\mathbf{x})) \otimes \mathbf{B}_g^{d_y}(b(\mathbf{y})) \tag{8.22}$$

It can be proved that the sum of all elements of the global basis vector $\mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y}))$ is equal to 1. That is, the stable local basis property of the splines is maintained.

Stable Basis Vector Theorem:
Given $|\mathbf{B}_g^{d_x}(b(\mathbf{x}))| = 1$, $|\mathbf{B}_g^{d_y}(b(\mathbf{y}))| = 1$ with $|\cdot|$ 1-norm and each element in both row vectors positive, it holds that $|\mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y}))| = 1$.
Proof:

Let the row vector $\mathbf{B}_g^{d_x}(b(\mathbf{x})) = \mathbf{X} = (a_1, a_2, \cdots, a_m)$, $\mathbf{B}_g^{d_y}(b(\mathbf{y})) = \mathbf{Y} = (b_1, b_2, \cdots, b_n)$ and $\mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y})) = \mathbf{Z} = \mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y})) = (\mathbf{Z_1}, \mathbf{Z_2}, \cdots, \mathbf{Z_m})$ for simplicity, then we have:

$$\begin{aligned}\mathbf{Z} &= \mathbf{X} \otimes \mathbf{Y} \\ &= (a_1, a_2, \cdots, a_m) \otimes (b_1, b_2, \cdots, b_n) \\ &= (a_1 \cdot \mathbf{Y}, a_2 \cdot \mathbf{Y}, \cdots, a_i \cdot \mathbf{Y}, \cdots, a_m \cdot \mathbf{Y})\end{aligned} \tag{8.23}$$

By defining $\mathbf{Z}_i = a_i \cdot \mathbf{Y}$ and $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_i, \cdots, \mathbf{Z}_m)$, it yields that:

$$\begin{aligned}|\mathbf{Z}| &= \sum_{i=1}^{m} |\mathbf{Z}_i| \\ &= \sum_{i=1}^{m} a_i \cdot |\mathbf{Y}| \\ &= \sum_{i=1}^{m} a_i \cdot 1 \\ &= |\mathbf{X}| = 1\end{aligned} \tag{8.24}$$

□

Correspondingly, the global polynomial function becomes:

$$p^t(\mathbf{x}, \mathbf{y}) := \mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y})) \cdot \mathbf{c}_{xy}, \;\; \mathbf{c}_{xy} \in \mathscr{R}^{J_x \hat{d}_x \cdot J_y \hat{d}_y}, \mathbf{x} \in t_{x_j} \; and \; \mathbf{y} \in t_{y_i} \tag{8.25}$$

The overall continuity matrix becomes:

$$\mathbf{H}_{xy} = \begin{bmatrix} \mathbf{H}_x \otimes \mathbb{I}_{J_y \hat{d}_y} \\ \mathbb{I}_{J_x \hat{d}_x} \otimes \mathbf{H}_y \end{bmatrix} \tag{8.26}$$

with $\mathbf{H}_{xy} \in \mathbb{R}^{\left(E_x \cdot R_x \cdot J_y \hat{d}_y + J_x \hat{d}_x \cdot E_y \cdot R_y\right) \times \left(J_x \cdot \hat{d}_x \cdot J_y \hat{d}_y\right)}$.

The directional derivatives become:

$$\frac{\partial p^t(\mathbf{x}, \mathbf{y})}{\partial x_i} = \frac{\partial \mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y}))}{\partial x_i} \cdot \mathbf{c}_{xy} \tag{8.27}$$

and

$$\frac{\partial p^t(\mathbf{x}, \mathbf{y}))}{\partial y_j} = \frac{\partial \mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y}))}{\partial y_j} \cdot \mathbf{c}_{xy} \tag{8.28}$$

where

$$\begin{aligned}\frac{\partial \mathbf{B}_{xy}^{d}(b(\mathbf{x}), b(\mathbf{y}))}{\partial x_i} &= \frac{\partial \mathbf{B}_{g}^{d_x}(b(\mathbf{x}))}{\partial x_i} \otimes \mathbf{B}_{g}^{d_y}(b(\mathbf{y})) + \mathbf{B}_{g}^{d_x}(b(\mathbf{x})) \otimes \frac{\partial \mathbf{B}_{g}^{d_y}(b(\mathbf{y}))}{\partial x_i} = \frac{\partial \mathbf{B}_{g}^{d_x}(b(\mathbf{x}))}{\partial x_i} \otimes \mathbf{B}_{g}^{d_y}(b(\mathbf{y})) \\ &= \left[\frac{d_x!}{(d_x - m_x)!} \mathbf{B}^{d_x - m_x}(b(\mathbf{x})) \cdot \mathbf{P}^{d_x, d_x - m_x}(a(\mathbf{u}_x)) \right] \otimes \mathbf{B}_{g}^{d_y}(b(\mathbf{y}))\end{aligned} \tag{8.29}$$

$$\frac{\partial \mathbf{B}_{xy}^{d}(b(\mathbf{x}),b(\mathbf{y}))}{\partial y_j}=\frac{\partial \mathbf{B}_{g}^{d_x}(b(\mathbf{x}))}{\partial y_j}\otimes\mathbf{B}_{g}^{d_y}(b(\mathbf{y}))+\mathbf{B}_{g}^{d_x}(b(\mathbf{x}))\otimes\frac{\partial \mathbf{B}_{g}^{d_y}(b(\mathbf{y}))}{\partial y_j}=\mathbf{B}_{g}^{d_x}(b(\mathbf{x}))\otimes\frac{\partial \mathbf{B}_{g}^{d_y}(b(\mathbf{y}))}{\partial y_j}$$
$$=\mathbf{B}_{g}^{d_x}(b(\mathbf{x}))\otimes\left[\frac{d_y!}{(d_y-m_y)!}\mathbf{B}^{d_y-m_y}\left(b\left(\mathbf{y}\right)\right)\cdot\mathbf{P}^{d_y,d_y-m_y}\left(a\left(\mathbf{u}_y\right)\right)\right] \tag{8.30}$$

with $\mathbf{P}^{d_x,d_x-m_x}(a(\mathbf{u}_x))$ and $\mathbf{P}^{d_y,d_y-m_y}\left(a\left(\mathbf{u}_y\right)\right)$ the de Castejau matrix, see [37].

In this chapter, only the first order directional derivative is required to construct the control effectiveness matrix. Therefore, m equals 1.

Based on Eqs. 8.29-8.30, considering dual partial derivatives, we can get:

$$\frac{\partial \mathbf{B}_{xy}^{d}(b(\mathbf{x}),b(\mathbf{y}))}{\partial x_i\cdot\partial y_j}=\frac{\partial \mathbf{B}_{g}^{d_x}(b(\mathbf{x}))}{\partial x_i}\otimes\frac{\partial \mathbf{B}_{g}^{d_y}(b(\mathbf{y}))}{\partial y_j}=\left[\frac{d_x!}{(d_x-m_x)!}\mathbf{B}^{d_x-m_x}(b(\mathbf{x}))\cdot\mathbf{P}^{d_x,d_x-m_x}(a(\mathbf{u}_x))\right]$$
$$\otimes\left[\frac{d_y!}{(d_y-m_y)!}\mathbf{B}^{d_y-m_y}\left(b\left(\mathbf{y}\right)\right)\cdot\mathbf{P}^{d_y,d_y-m_y}\left(a\left(\mathbf{u}_y\right)\right)\right] \tag{8.31}$$

Using Eq. 8.31, the cross partial derivatives of $p^t(\mathbf{x},\mathbf{y})$ defined in Eq. 8.25 can be derived as follows:

$$\frac{\partial p^t(\mathbf{x},\mathbf{y})}{\partial x_i\partial y_j}=\frac{\partial \mathbf{B}_{xy}^{d}(b(\mathbf{x}),b(\mathbf{y}))}{\partial x_i\partial y_j}\cdot\mathbf{c}_{xy} \tag{8.32}$$

Eq. 8.32 and Eq. 8.31 together show that the partial derivative of $p^t(\mathbf{x},\mathbf{y})$ can be treated differently in different dimensions. In contrast, standard simplex B-splines can only obtain homogeneous directional derivatives in each dimension.

It also should be noted that the continuity orders defined for different dimensions are maintained during the process of combination from Eqs. 8.20-8.21 to Eq. 8.25. For example, if $p^t(\mathbf{x})$ is second order differentiable with respect to x_i and $p^t(\mathbf{y})$ is first order differentiable with respect to y_j, then $p^t(\mathbf{x},\mathbf{y})$ is second order differentiable with respect to x_i and first order differentiable with respect to y_j.

Eq. 8.31 shows vector $\mathbf{y}$ can be treated differently from $\mathbf{x}$. More generally, TP simplex B-splines enable the input space to be divided into multiple segments, and treat different segments differently in terms of selection of polynomial orders or smoothness requirements. For example, the overall input vector $\mathbf{u}$ can be divided as: $\mathbf{u}=[\mathbf{x},\mathbf{y},\mathbf{z}]$. This leads to the following expression:

$$\frac{\partial \mathbf{B}_{xyz}^{d}\left(b\left(\mathbf{x}\right),b\left(\mathbf{y}\right),b\left(\mathbf{z}\right)\right)}{\partial y_j}=\mathbf{B}_{g}^{d_x}(b(\mathbf{x}))\otimes\frac{\partial \mathbf{B}_{g}^{d_y}\left(b(\mathbf{y})\right)}{\partial y_j}\otimes\mathbf{B}_{g}^{d_z}(b(\mathbf{z}))$$
$$=\mathbf{B}_{g}^{d_x}(b(\mathbf{x}))\otimes\left[\frac{d_y!}{(d_y-m_y)!}\mathbf{B}^{d_y-m_y}\left(b\left(\mathbf{y}\right)\right)\cdot\mathbf{P}^{d_y,d_y-m_y}\left(a\left(\mathbf{u}_y\right)\right)\right]\otimes\mathbf{B}_{g}^{d_z}(b(\mathbf{z})) \tag{8.33}$$

where $\mathbf{B}_{xyz}^{d}\left(b\left(\mathbf{x}\right),b\left(\mathbf{y}\right),b\left(\mathbf{z}\right)\right)$ is defined in the same way as in Eq. 8.22.

8.4.4. COMPUTATIONAL COMPLEXITY OF TP SIMPLEX SPLINES

The TP simplex B-splines presented in Section 8.4.3 provide users with more options during model structure selection when using simplex B-splines. Taking the identification of C_l as an example, it is assumed that we lack *a priori* knowledge of the structure of the spline function concerning the arguments α, β and $\bar{p}$. Specifically, the detailed model structure as shown in [101, 157] is assumed unknown for the sake of our argument, and we only know that C_l is affine to δ_a. Actually, this assumption is realistic according to [114]. In this case, the model structure can be further simplified according to the *a priori* knowledge. Instead of constructing a four dimensional spline function $s_d^m(\alpha,\beta,\bar{p},\delta_a)$, we can construct an equivalent model using the tensor-product structure by setting $\mathbf{x}=[\alpha,\beta,\bar{p}]$ and $\mathbf{y}=\delta_a$ in Eq. 8.20 and Eq. 8.21.

Compared to a standard simplex B-spline model, the tensor-product simplex B-splines require fewer B-coefficients than the former and therefore lead to lower computational load in a real-time implementation. According to [125] and [73][p.110], a Delaunay triangulation produces $\mathcal{O}\left\{n_v^{d_f}\right\}$ simplices, with n_v the number of vertices and d_f the dimension of the functional space.

The number of the sub domains produced by the tensor-product simplex B-splines is compared with that of the standard simplex B-splines, and the results are listed in Table 8.1. In this table, x, y , z and k represent the number of partitioning knots in each dimension of the arguments, and n, n_1, n_2 and n_3 denote the dimension of the basis function.

Table 8.1: Total number of simplices produced by TP simplex splines and standard simplex splines when using standard Delaunay triangulation method

algorithms	case one $n=3,n_1=2,n_2=1$	case two $n=4,n_1=n-1,n_2=1$	case three $n=4,n_1=n-2,n_2=1,n_3=1$
standard simplex splines, $\mathcal{O}$	$(x\cdot y\cdot z)^{(\frac{n}{2})}$	$(x\cdot y\cdot z\cdot k)^{(\frac{n}{2})}$	$(x\cdot y\cdot z\cdot k)^{(\frac{n}{2})}$
TP simplex splines, $\mathcal{O}$	$(x\cdot y)^{(\frac{n-1}{2})}\cdot(z-1)$	$(x\cdot y\cdot z)^{(\frac{n-1}{2})}\cdot(k-1)$	$(x\cdot y)^{(\frac{n-2}{2})}\cdot(z-1)\cdot(k-1)$
ratio $\frac{\mathcal{O}(\text{std})}{\mathcal{O}(\text{tps})}$	$\geq 3^{\frac{3}{2}}\approx 5.20$	$\geq 8\sqrt{2}\approx 11.31$	≥ 64

As shown in the first column of Table 8.1, the ratio of the number of the sub domains is:

$$\frac{(x\cdot y\cdot z)^{(\frac{n}{2})}}{(x\cdot y)^{(\frac{n-1}{2})}\cdot(z-1)}=\sqrt{xy}\cdot\frac{z^{\frac{n}{2}}}{(z-1)} \tag{8.34}$$

Given that $x\geq 2$, $y\geq 2$, $z\geq 2$ and $n=3$, the partial derivative of Eq. 8.34 with regard to z is:

$$\begin{aligned}\sqrt{xy}\cdot\frac{z^{\frac{n}{2}}}{(z-1)}|_{dz} &= \sqrt{xy}\cdot\frac{z^{\frac{3}{2}}}{(z-1)}|_{dz}\\ &= \frac{3}{2}\frac{z^{\frac{1}{2}}}{z-1}-\frac{z^{\frac{3}{2}}}{(z-1)^2}\\ &= \frac{z^{\frac{1}{2}}}{z-1}\cdot\left(\frac{3}{2}-\frac{z}{z-1}\right)\\ &= \frac{z^{\frac{1}{2}}}{z-1}\cdot\left(\frac{3}{2}-1-\frac{1}{z-1}\right)\end{aligned} \tag{8.35}$$

$\sqrt{xy}\cdot\frac{z^{\frac{n}{2}}}{(z-1)}|_{dz}=0$ when $z=3$, and Eq. 8.34 has a minimum:

$$\begin{aligned}\sqrt{xy}\cdot\frac{z^{\frac{n}{2}}}{(z-1)} &= \sqrt{xy}\cdot\frac{z^{\frac{3}{2}}}{(z-1)}\\ &\geq \sqrt{xy}\cdot\frac{3^{\frac{3}{2}}}{(3-1)}\\ &\geq \sqrt{2\cdot 2}\cdot\frac{3^{\frac{3}{2}}}{(3-1)}\\ &\geq 3^{\frac{3}{2}}\end{aligned} \tag{8.36}$$

It is proved that tensor-product simplex B-splines produce a lower number of sub domains than the standard simplex B-splines when using the standard Delaunay triangulation method. Similarly, the same conclusion can be drawn for the remaining two cases in Table 8.1. To summarize, the tensor-product simplex B-splines produce a lower number of sub domains, i.e., simplices, than the standard simplex B-splines.

8

Remarks: Recall that the local per-simplex basis regression vector produced by tensor-product simplex B-splines always contains a lower number of polynomial terms than that produced by standard simplex B-splines, see Section 8.4.1. To summarize, combining the conclusions from both Section 8.4.1 and Section 8.4.4, we can draw the conclusion that the tensor-product simplex B-splines require a lower number of global B-coefficients than the standard simplex B-splines. Therefore, it is more likely that the former causes a lower computational load than the latter in the real-time implementation.

8.5. VALIDATION AND COMPARISON WITH STANDARD SIMPLEX SPLINES

8.5.1. F-16 MODEL STRUCTURE SELECTION WITH COMPLETE A PRIORI KNOWLEDGE

The estimation of the non-dimensional moment coefficients is investigated in this chapter. According to the complete *a priori* knowledge from [114], the following model

structures can be assumed for the moment coefficients of an F-16 aircraft:

$$\begin{aligned} &C_l\left(\alpha,\beta,\tilde{p},\tilde{r},\delta_a,\delta_r,\delta_{lef}\right) = C_l\left(\alpha,\beta\right) \\ &+ C_{l_{\delta_{lef}}}\left(\alpha,\beta\right)\delta_{lef} + C_{l_{\delta_a}}\left(\alpha,\beta\right)\delta_a + C_{l_{\delta_r}}\left(\alpha,\beta\right)\delta_r \\ &+ C_{l_r}\left(\alpha\right)\tilde{r} + C_{l_r\delta_{lef}}\left(\alpha\right)\delta_{lef}\tilde{r} + C_{l_p}\left(\alpha\right)\tilde{p} + C_{l_p\delta_{lef}}\left(\alpha\right)\delta_{lef}\tilde{p} \end{aligned} \tag{8.37}$$

with $\bar{p} = \frac{pb}{2V}$ and $\bar{r} = \frac{rb}{2V}$.

$$\begin{aligned} &C_m\left(\alpha,\beta,\tilde{q},\delta_e,\delta_{lef}\right) = C_m\left(\alpha,\beta,\delta_e\right) + C_{m_{\delta_{lef}}}\left(\alpha,\beta\right)\delta_{lef} \\ &+ C_{m_q}\left(\alpha\right)\tilde{q} + C_{m_q\delta_{lef}}\left(\alpha\right)\delta_{lef}\tilde{q} \end{aligned} \tag{8.38}$$

with $\bar{q} = \frac{q\bar{c}}{V}$.

$$\begin{aligned} &C_n\left(\alpha,\beta,\tilde{p},\tilde{r},\delta_a,\delta_r,\delta_{lef}\right) = C_n\left(\alpha,\beta\right) \\ &+ C_{n_{\delta_{lef}}}\left(\alpha,\beta\right)\delta_{lef} + C_{n_{\delta_a}}\left(\alpha,\beta\right)\delta_a + C_{n_{\delta_r}}\left(\alpha,\beta\right)\delta_r \\ &+ C_{n_r}\left(\alpha\right)\tilde{r} + C_{n_r\delta_{lef}}\left(\alpha\right)\delta_{lef}\tilde{r} + C_{n_p}\left(\alpha\right)\tilde{p} + C_{n_p\delta_{lef}}\left(\alpha\right)\delta_{lef}\tilde{p} \end{aligned} \tag{8.39}$$

Accordingly, each subfunction in the above expressions can be identified using a spline function. Eqs. 8.37-8.39 become:

$$\begin{aligned} &C_l\left(\alpha,\beta,\tilde{p},\tilde{r},\delta_a,\delta_r,\delta_{lef}\right) = s_{11}\left(\alpha,\beta\right) \\ &+ s_{12}\left(\alpha,\beta\right)\delta_{lef} + s_{13}\left(\alpha,\beta\right)\delta_a + s_{14}\left(\alpha,\beta\right)\delta_r \\ &+ s_{15}\left(\alpha\right)\tilde{r} + s_{16}\left(\alpha\right)\delta_{lef}\tilde{r} + s_{17}\left(\alpha\right)\tilde{p} + s_{18}\left(\alpha\right)\delta_{lef}\tilde{p} \end{aligned} \tag{8.40}$$

$$\begin{aligned} &C_m\left(\alpha,\beta,\tilde{q},\delta_e,\delta_{lef}\right) = s_{21}\left(\alpha,\beta,\delta_e\right) + s_{22}\left(\alpha,\beta\right)\delta_{lef} \\ &+ s_{23}\left(\alpha\right)\tilde{q} + s_{24}\left(\alpha\right)\delta_{lef}\tilde{q} \end{aligned} \tag{8.41}$$

$$\begin{aligned} &C_n\left(\alpha,\beta,\tilde{p},\tilde{r},\delta_a,\delta_r,\delta_{lef}\right) = s_{31}\left(\alpha,\beta\right) \\ &+ s_{32}\left(\alpha,\beta\right)\delta_{lef} + s_{33}\left(\alpha,\beta\right)\delta_a + s_{34}\left(\alpha,\beta\right)\delta_r \\ &+ s_{35}\left(\alpha\right)\tilde{r} + s_{36}\left(\alpha\right)\delta_{lef}\tilde{r} + s_{37}\left(\alpha\right)\tilde{p} + s_{38}\left(\alpha\right)\delta_{lef}\tilde{p} \end{aligned} \tag{8.42}$$

8.5.2. F-16 MODEL STRUCTURE SELECTION WITH LIMITED A PRIORI KNOWLEDGE

According to Section 8.5.1, each subfunction in Eqs. 8.37-8.39 is at most of dimension 3, and the total number of the unknown parameters is relatively low. In this particular case, TP simplex B-splines have no chance to show its advantage over standard simplex B-splines in terms of computational efficiency. However, this kind of *complete a priori* knowledge on model structures usually does not apply to most of other nonlinear

systems, e.g., aircraft systems. In order to demonstrate the advantages of TP simplex B-splines using the data set from an F-16 aircraft, it is assumed in this chapter that only very limited *a priori* model knowledge is available, see the assumption below.

Assumption on limited *a priori* model structure knowledge:
The dynamics of a nonlinear system, which includes but is not only limited to an F-16 aircraft system, can be captured using a simplex B-spline model structure with the following property: the polynomial orders required for certain inputs are unnecessary to be as high as other inputs. More precisely, it is assumed for an F-16 model that the required polynomial orders for the control surface related inputs, i.e., $\delta_a, \delta_e, \delta_r$, are lower than those for other dimensions of inputs. That is, the control inputs of the system are less coupled with the system states.

Under the abovementioned assumption, the spline model for the 3 dimensionless moment coefficients could be chosen as follows:

$$\mathrm{C_m}\left(\alpha,\beta,\bar{q},\delta_e,\delta_{\mathrm{lef}},\delta_T\right)=\left[\mathbf{B}_{gm}^{d_{1m}}\left(\alpha,\beta,\bar{q}\right)\otimes\mathbf{B}_{gm}^{d_{2m}}\left(\delta_e\right)\otimes\mathbf{B}_{gm}^{d_{3m}}\left(\delta_{\mathrm{lef}}\right),\delta_T\right]\cdot\mathbf{c}_{gm} \tag{8.43}$$

$$\mathrm{C_l}\left(\alpha,\beta,\bar{p},\bar{r},\delta_a,\delta_r\right)=\left[\mathbf{B}_{gl}^{d_{1l}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\mathbf{B}_{gl}^{d_{2l}}\left(\delta_a\right)\otimes\mathbf{B}_{gl}^{d_{3l}}\left(\delta_r\right)\right]\cdot\mathbf{c}_{gl} \tag{8.44}$$

$$\mathrm{C_n}\left(\alpha,\beta,\bar{p},\bar{r},\delta_r,\delta_a\right)=\left[\mathbf{B}_{gn}^{d_{1n}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\mathbf{B}_{gn}^{d_{2n}}\left(\delta_r\right)\otimes\mathbf{B}_{gn}^{d_{3n}}\left(\delta_a\right)\right]\cdot\mathbf{c}_{gn} \tag{8.45}$$

For the purpose of control allocation, the control derivatives are required, and they can be obtained through calculating the partial derivatives of a TP simplex B-spline function. Combining Eqs. 8.43-8.45 with Eqs. 8.29-8.30, the control derivatives can be calculated as follows:

$$\begin{aligned}\boldsymbol{D}_{\delta_e}\cdot\mathrm{C_m}\left(\alpha,\beta,\bar{q},\delta_e,\delta_{\mathrm{lef}},\delta_T\right)&=\left[\mathbf{B}_{gm}^{d_{1m}}\left(\alpha,\beta,\bar{q}\right)\otimes\frac{\partial\mathbf{B}_{g}^{d_{2m}}\left(b(\delta_e)\right)}{\partial\delta_e}\otimes\mathbf{B}_{gm}^{d_{3m}}\left(\delta_{\mathrm{lef}}\right),\delta_T\right]\cdot\mathbf{c}_{gm}\\&=\left[\mathbf{B}_{gm}^{d_{1m}}\left(\alpha,\beta,\bar{q}\right)\otimes\left[\frac{d_{2m}!}{(d_{2m}-m_{2m})!}\mathbf{B}^{d_{2m}-m_{2m}}\left(b\left(\delta_e\right)\right)\cdot\mathbf{P}^{d_{2m},d_{2m}-m_{2m}}\left(a\left(\mathbf{u}_{\delta_e}\right)\right)\right]\otimes\mathbf{B}_{gm}^{d_{3m}}\left(\delta_{\mathrm{lef}}\right),\delta_T\right]\cdot\mathbf{c}_{gm}\end{aligned} \tag{8.46}$$

with $\boldsymbol{D}$ the partial derivative operator.

$$\begin{aligned}\boldsymbol{D}_{\delta_a}\cdot\mathrm{C_l}\left(\alpha,\beta,\bar{p},\bar{r},\delta_a,\delta_r\right)&=\left[\mathbf{B}_{gl}^{d_{1l}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\frac{\partial\mathbf{B}_{g}^{d_{2l}}\left(b(\delta_a)\right)}{\partial\delta_a}\otimes\mathbf{B}_{gl}^{d_{3l}}\left(\delta_r\right)\right]\cdot\mathbf{c}_{gl}\\&=\left[\mathbf{B}_{gl}^{d_{1l}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\left[\frac{d_{2l}!}{(d_{2l}-m_{2l})!}\mathbf{B}^{d_{2l}-m_{2l}}\left(b\left(\delta_a\right)\right)\cdot\mathbf{P}^{d_{2l},d_{2l}-m_{2l}}\left(a\left(\mathbf{u}_{\delta_a}\right)\right)\right]\otimes\mathbf{B}_{gl}^{d_{3l}}\left(\delta_r\right)\right]\cdot\mathbf{c}_{gl}\end{aligned} \tag{8.47}$$

$$\begin{aligned}\boldsymbol{D}_{\delta_r}\cdot\mathrm{C_l}\left(\alpha,\beta,\bar{p},\bar{r},\delta_a,\delta_r\right)&=\left[\mathbf{B}_{gl}^{d_{1l}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\mathbf{B}_{gl}^{d_{2l}}\left(\delta_a\right)\otimes\frac{\partial\mathbf{B}_{g}^{d_{3l}}\left(b(\delta_r)\right)}{\partial\delta_r}\right]\cdot\mathbf{c}_{gl}\\&=\left[\mathbf{B}_{gl}^{d_{1l}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\mathbf{B}_{gl}^{d_{2l}}\left(\delta_a\right)\otimes\left[\frac{d_{3l}!}{(d_{3l}-m_{3l})!}\mathbf{B}^{d_{3l}-m_{3l}}\left(b\left(\delta_r\right)\right)\cdot\mathbf{P}^{d_{3l},d_{3l}-m_{3l}}\left(a\left(\mathbf{u}_{\delta_r}\right)\right)\right]\right]\cdot\mathbf{c}_{gl}\end{aligned} \tag{8.48}$$

$$\boldsymbol{D}_{\delta_r}\cdot \mathrm{Cn}\left(\alpha,\beta,\bar{p},\bar{r},\delta_r,\delta_a\right)=\left[\mathbf{B}_{gn}^{d_{1n}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\frac{\partial\mathbf{B}_{g}^{d_{2n}}\left(b(\delta_r)\right)}{\partial\delta_r}\otimes\mathbf{B}_{gn}^{d_{3n}}\left(\delta_a\right)\right]\cdot\mathbf{c}_{gn}$$
$$=\left[\mathbf{B}_{gn}^{d_{1n}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\left[\frac{d_{2n}!}{(d_{2n}-m_{2n})!}\mathbf{B}^{d_{2n}-m_{2n}}\left(b(\delta_r)\right)\cdot\mathbf{P}^{d_{2n},d_{2n}-m_{2n}}\left(a\left(\mathbf{u}_{\delta_r}\right)\right)\right]\otimes\mathbf{B}_{gn}^{d_{3n}}\left(\delta_a\right)\right]\cdot\mathbf{c}_{gn} \tag{8.49}$$

$$\boldsymbol{D}_{\delta_a}\cdot \mathrm{Cn}\left(\alpha,\beta,\bar{p},\bar{r},\delta_r,\delta_a\right)=\left[\mathbf{B}_{gn}^{d_{1n}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\mathbf{B}_{gn}^{d_{2n}}\left(\delta_r\right)\otimes\frac{\partial\mathbf{B}_{g}^{d_{3n}}\left(b(\delta_a)\right)}{\partial\delta_a}\right]\cdot\mathbf{c}_{gn}$$
$$=\left[\mathbf{B}_{gn}^{d_{1n}}\left(\alpha,\beta,\bar{p},\bar{r}\right)\otimes\mathbf{B}_{gn}^{d_{2n}}\left(\delta_r\right)\otimes\left[\frac{d_{3n}!}{(d_{3n}-m_{3n})!}\mathbf{B}^{d_{3n}-m_{3n}}\left(b(\delta_a)\right)\cdot\mathbf{P}^{d_{3n},d_{3n}-m_{3n}}\left(a\left(\mathbf{u}_{\delta_a}\right)\right)\right]\right]\cdot\mathbf{c}_{gn} \tag{8.50}$$

As shown in Eqs. 8.43-8.45, the *a priori* model information can be easily taken into account by TP simplex B-spline model structure, and the control surface deflections are treated differently from other aircraft states.

In view of the computational complexity of control allocation, TP simplex B-splines have some advantages over standard simplex B-splines. As shown in [157], the identified spline function needs to be parameterized, and the gradient, perhaps also the Hessian matrix, of the polynomial basis vector needs to be calculated for each per-simplex local model at each optimization step during control allocation. Using standard simplex B-splines, a large number of simplices will be involved in the control allocation optimization process at each sample step. This is because the control input is also involved in the partitioning of the aircraft state space, e.g., a three dimensional space defined on α, β and $\bar{q}$. Once the control input is adjusted, the current evaluation point will probably enters another simplex.

In contrast, as indicated by the results shown in Table 8.1, much less simplices are involved in an optimization process during control allocation when using TP simplex B-splines. This is because the control inputs are assumed less coupled with other aircraft states in TP simplex B-splines. The control input is not involved in the partitioning of defined on the aircraft states. To summarize, the computational load of the control allocation solver caused by calculating a large number of per-simplex local optimum can be relieved.

With the function inputs determined by Eqs. 8.43-8.45, model structure selection for system identification problem using the simplex B-splines is reduced into the selection of triangulation vertices, per-simplex polynomial orders and smoothness orders. The working ranges of the control inputs and the state variables of an F-16 aircraft are shown in Eq. 8.51.

$$\begin{array}{llll} \alpha & :=[-10,45]\,[\mathrm{deg}] & \delta_e := & [-25,25]\,[\mathrm{deg}] \\ \beta & :=[-30,30]\,[\mathrm{deg}] & \delta_a := & [-21.5,21.5]\,[\mathrm{deg}] \\ q & :=[-30,30]\,[\mathrm{deg}/s] & \delta_r := & [-30,30]\,[\mathrm{deg}] \\ p & :=[-30,30]\,[\mathrm{deg}/s] & \delta_{lef} := & [0,25]\,[\mathrm{deg}] \\ r & :=[-30,30]\,[\mathrm{deg}/s] & V_{\mathrm{TAS}} := & [50,300]\,[\mathrm{m}/s] \end{array} \tag{8.51}$$

8.5.3. SIMULATION RESULTS COMPARED TO STANDARD SIMPLEX B-SPLINES

This section is focused on demonstrating the benefits of using the TP simplex B-spline model structure compared to the standard simplex B-spline structure. Specifically, the approximation power of the TP simplex B-splines is investigated for different polynomial degrees, and the results are compared with those from the standard simplex splines. However, the data sets used in this section are limited to a 3-D data set and a 5-D data set. In contrast, the focus of Section 8.6 is shifted from the validation of the TP simplex spline model to its application to an F-16 aircraft. More validation results will be given in Section 8.6 where the approximation accuracy for the roll and yaw moment coefficients of an F-16 aircraft are also investigated. Keep in mind that different data sets and triangulations will be used in Section 8.6. The validation results are extended to all three moment coefficients of an F-16 aircraft in Section 8.6, and the polynomial order is chosen at a fixed number. Two scenarios are used to demonstrate the approximation power of the TP simplex B-splines in this section:

Scenario 1: The term $C_{\mathrm{m}}\left(\alpha,\beta,\delta_e\right)$ shown in Eq. 8.38 is approximated using a 3-D data set generated using the wind tunnel data of an F-16 aircraft. The $s_{\mathrm{tp}}\left(\alpha,\beta,\delta_e\right)-s_{\mathrm{tp}}\left(2,1\right)$ and $s_{\mathrm{std}}\left(\alpha,\beta,\delta_e\right)$ model structures are used for the TP simplex and the standard simplex splines, respectively.

Scenario 2: The pitch moment coefficient C_{m} is approximated from a 5-D data set generated using the wind tunnel data of an F-16 aircraft. The $s_{\mathrm{tp}}\left(\alpha,\beta,\tilde{q},\delta_e,\delta_{lef}\right)-s_{\mathrm{tp}}\left(3,1,1\right)$ and $s_{\mathrm{std}}\left(\alpha,\beta,\tilde{q},\delta_e,\delta_{lef}\right)$ model structures are used for the TP simplex and the standard simplex splines, respectively.

For both scenarios 1 and 2, the training data set consists 12000 scattered data points, and the test data set consists of 4000 data points.

It should be noted that the term $C_{\mathrm{m}}\left(\alpha,\beta,\delta_e\right)$ shown in Eq. 8.38 is only a subfunction of C_{m}, therefore, it is different from C_{m}. To indicate the difference, $C_{\mathrm{m}}\left(\alpha,\beta,\delta_e\right)$ will be written as $C^*_{\mathrm{m}}\left(\alpha,\beta,\delta_e\right)$ in the remainder of this chapter. In scenario 1, the grids chosen to build a triangulation are: $[-10,10,45]$, $[-30,30]$, $[-25,0,25]$. An $s_{\mathrm{tp}}\left(\alpha,\beta,\delta_e\right)-s_{\mathrm{tp}}\left(2,1\right)$ TP simplex B-spline structure is used after assuming that the third dimension δ_e is less correlated to the other two dimension of inputs. The individual dimensions for the spline function division and the smoothness order are chosen as $n_1=2, m_1=1$, $n_2=1, m_2=1$. In addition, as a comparison, a 3-D function using standard simplex B-splines is approximated without splitting the entire function into multiple sub structures, where $n=3, m=1$.

The Solver for Spline Model Identification

The function approximation problem shown by Eqs. 4.8 and 4.13 is rewritten as follows:

$$\begin{aligned}\mathbf{Y}&=\mathbf{X}\cdot\mathbf{c}+\xi\\ \mathbf{Hc}&=0\end{aligned}\tag{8.52}$$

where $\mathbf{X}$ is the regression matrix with each row vector built using $\mathbf{B}_g^d(b(\mathbf{x}))$, $\mathbf{Y}$ is the observation vector. It is assumed that $\boldsymbol{E}(\xi)=0$, $\boldsymbol{Cov}(\xi)=\sum$ with $\sum$ the residual covariance matrix.

The cost function of the spline function approximation problem is:

$$\boldsymbol{J}(\mathbf{c})=\frac{1}{2}(\mathbf{Y}-\mathbf{X}\cdot\mathbf{c})^{\top}\sum^{-1}(\mathbf{Y}-\mathbf{X}\cdot\mathbf{c}) \tag{8.53}$$

The optimization problem is formulated as follows:

$$\min_{\mathbf{c}}\boldsymbol{J}(\mathbf{c}),\ subject\ to\ \mathbf{Hc}=0 \tag{8.54}$$

Using Lagrange multipliers this problem can be formulated as a Karush-Kuhn-Tucker (KKT) system, and an solution can be efficiently derived either using an iterative solver developed by Awanou and Lai et al. [9, 10] or using a substitution-based solver, see Chapter 3. The first solver is batch type method, the second solver suits for online implementation.

Table 8.2: Approximation power comparison between TP simplex B-splines and standard simplex B-splines, 3-D $C_{\mathrm{m}}^{*}(\alpha,\beta,\delta_e)$

TP simplex B-splines, $s_{\mathrm{tp}}(2,1)$,$\mathscr{T}_4$,$\mathscr{T}_2$				standard simplex B-splines, $s_{\mathrm{std}}(3)$, $\mathscr{T}_{24}$			
polynomial order	$n_1=2,m_1=1,n_2=1,m_2=1$			polynomial order	$n=3,m=1$		
	coef. num.	RMSE	max error		coef. num.	RMSE	max error
$d_1=2,d_2=2$	144	0.0189	0.0987	$d=2$	240	0.0238	0.1142
$d_1=3,d_2=3$	320	0.0139	0.0840	$d=3$	480	0.0153	0.0964
$d_1=4,d_2=3$	480	0.0110	0.0596	$d=4$	840	0.0116	0.0727
$d_1=4,d_2=4$	600	0.0108	0.0577	$d=4$	840	0.0116	0.0727
$d_1=5,d_2=5$	1008	0.0080	0.0603	$d=5$	1344	0.0082	0.0372

According to the results shown in Table 8.2, TP simplex B-splines can lead to a higher approximation accuracy than the standard simplex B-splines using less B-coefficients in the sense that the former yields a lower RMSE.

In scenario 2, the grids are: $[-10,10,45]$, $[-30,30]$, $[-25,25]$, $[0,25]$, $[-90,90]$. An $s_{\mathrm{tp}}(\alpha,\beta,\tilde{q},\delta_e,\delta_{lef})-s_{\mathrm{tp}}(3,1,1)$ structure is used when using TP simplex B-splines after assuming that the fourth and the fifth dimensions are less correlated with the other three dimension of inputs, $n_1=3$, $d_1=3$, $m_1=1$, $n_2=1$, $d_2=3$, $m_2=1$, $n_3=1$, $d_3=3$, $m_3=1$. In addition, as a comparison, a 3-D function using standard simplex B-splines is approximated without splitting the entire function into multiple sub structures, $n=5$, $d=3$, $m=1$. According to Table 8.3, the number of simplices increases very fast with

Table 8.3: Approximation power comparison between TP simplex B-splines and standard simplex B-splines, 5-D C_{m}

TP simplex B-splines, $s_{\mathrm{tp}}(3,1,1)$,$\mathscr{T}_6$,$\mathscr{T}_1$,$\mathscr{T}_1$				standard simplex B-splines, $s_{\mathrm{std}}(5)$,$\mathscr{T}_{96}$			
polynomial order	$n_1=3,m_1=1,n_2=1,m_2=1,n_3=1,m_3=1$			polynomial order	$n=5,m=1$		
	coef. num.	RMSE	max error		coef. num.	RMSE	max error
$d_1=d_2=d_3=2$	540	0.0286	0.1514	$d=2$	2016	0.0342	0.1690
$d_1=d_2=d_3=3$	1920	0.0214	0.1519	$d=3$	5376	0.0253	0.1166
$d_1=4,d_2=d_3=3$	3360	0.0177	0.1177	$d=3$	5376	0.0253	0.1166

the increase of dimensions. Using standard simplex B-splines, a triangulation contains

96 simplices when the spline function is dimension 5. The results shown in Table 8.3 support the conclusion presented in Section 8.4.4 that TP simplex B-splines usually yield much less simplices than standard simplex B-spline model structure. Similar to the results shown in Table 8.2, results shown in Table 8.3 also show that TP simplex B-splines have the potential to result in higher approximation accuracy using less B-coefficients than standard simplex B-spline model structure.

8.6. APPLICATION TO AN F-16 FOR MOMENT COEFFICIENT AND CONTROL DERIVATIVE ESTIMATION

The model identification method which uses TP simplex B-spline model structure is further validated using two simulated flight-trajectory data sets generated using a high-fidelity nonlinear F-16 model, see [114]. All three dimensionless moment coefficients are estimated using online data points available in a flight mission. The trim condition of the aircraft is: $V_{\text{TAS}} = 152.40$ m/s, $h_{\text{e}} = 4.572$ km. To obtain the first data set, the modified *3211* inputs shown in Figure 8.5 were applied to the open-loop F-16 model. To obtain the second data set, a sequence of angular commands was applied to a closed-loop F-16 model, which is equipped with an adaptive nonlinear dynamic inversion (ANDI) controller, see Appendix G. Each of the data sets mentioned above contained 3000 data points. The work presented in this section is focused on how well the TP simplex B-spline approach is suited for estimating all three nondimensional moment coefficients using the real-time data available in a simulated flight. In addition, attention is also paid on how to calculate the directional (control) derivatives, which constitute the control allocation matrix.

8.6.1. TRIANGULATIONS AND POLYNOMIAL ORDER DETERMINATION

The overall model structures of the spline models of C_{m}, C_{l} and C_{n} are shown in Eqs. 8.43-8.45. The triangulation of the simplex B-splines is constructed using the vertices shown in Eq. 8.51. The number of simplices in each triangulation is depicted in Table 8.4.

Table 8.4: The number of simplices contained by each triangulations

moments	C_{m}	C_{l}	C_{n}
triangulation	$\mathcal{T}_{1m},\mathcal{T}_{2m},\mathcal{T}_{3m}$	$\mathcal{T}_{1l},\mathcal{T}_{2l},\mathcal{T}_{3l}$	$\mathcal{T}_{1n},\mathcal{T}_{2n},\mathcal{T}_{3n}$
simplex number	6,1,1	21,1,1	21,1,1

The selected polynomial orders for the non-dimensional moment coefficients are listed in Table 8.5. Following the setups in Table 8.4 and Table 8.5, the overall B-

Table 8.5: Selection of polynomial orders and smoothness orders for the estimation of moment coefficients

moments	C_{m}	C_{l}	C_{n}
polynomial order	$d_{1m}=3,d_{2m}=3,d_{3m}=2$	$d_{1l}=2,d_{2l}=3,d_{3l}=3$	$d_{1n}=2,d_{2n}=3,d_{3n}=3$
smoothness order	$m_{1m}=0,m_{2m}=0,m_{3m}=0$	$m_{1l}=0,m_{2l}=0,m_{3l}=0$	$m_{1n}=0,m_{2n}=0,m_{3n}=0$

coefficient vectors $\mathbf{c}_{gm}$, $\mathbf{c}_{gl}$ and $\mathbf{c}_{gn}$, shown in Eqs. 8.43-8.45, have 1441, 5040 and

5040 elements, respectively. Then, by using the substitution based speedup strategy presented in [138], the lengths of the unknown vectors are reduced to 769, 1264 and 1264, respectively. It can be seen that the new speedup approach taken from [138] greatly reduces the scale of the original parameter identification problems.

8.6.2. VALIDATION RESULTS USING A 3211 OPEN-LOOP DATA SET

As mentioned at the beginning of this section, the changing history of the control inputs during collecting the first data set is illustrated in Figure 8.5. The 3211 type aileron, elevator and rudder inputs are applied to the open-loop nonlinear F-16 aircraft model sequentially. As can be seen in Figure 8.5, the magnitude of the 3211 type aileron inputs has been slightly modified to avoid a roll angle which is larger than 90 deg.

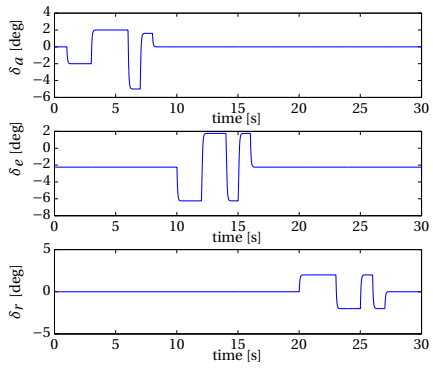


Figure 8.5: Excitation inputs to the open-loop model, 3211.

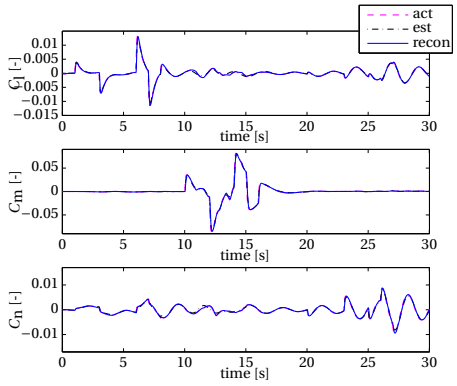


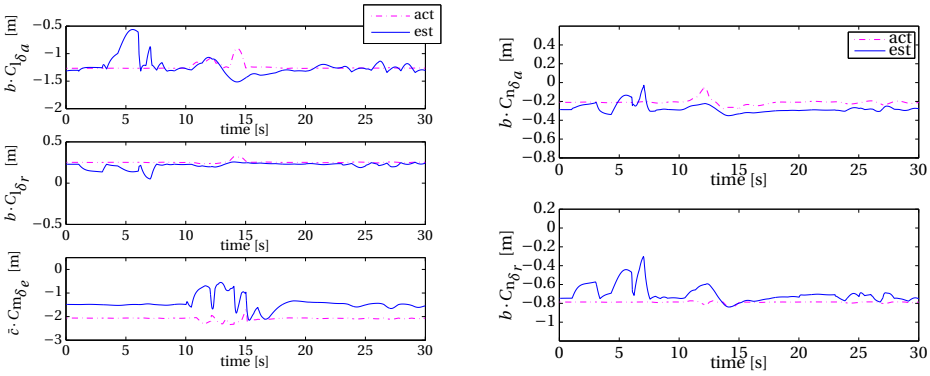
Figure 8.6: Comparison of the actual, reconstructed and estimated values of the non-dimensional moment coefficients.

The first data set is used as the training data set to demonstrate the applicability of the tensor-product simplex B-splines. The functional approximation power of the tensor-product simplex B-splines is depicted in Figure 8.6. In these figures, 'act', 'recon' and 'est' denote 'actual', 'reconstructed' and 'estimated', respectively. The actual value is constructed from wind-tunnel data of the F-16 aircraft model and interpolation, see [114]. The reconstructed values are the moment coefficients constructed from the measured states using Eqs. 8.3-8.5.

As shown in Figure 8.6, the output value of the spline function, i.e., C_l, stays very close to the actual and reconstructed value of C_l. The fitting error is negligible. Comparing the results shown in the period of 25-30 seconds with those shown in the period of 7-10 seconds, it can be found that the roll moment in the second period is larger on average than that in the first period. In Figure 8.6, small tracking errors only appear around the 14^{th} and 15^{th} seconds. Similar to the fitting results for C_l, the fitting errors of C_n shown in Figure 8.6 are also negligible. To summarize, the TP simplex B-splines with its structure shown in Eqs. 8.43-8.45 have shown adequate approximation power in fitting the moment coefficients of an F-16 aircraft.

As presented in [87], the control effectiveness matrix, which is comprised of control

derivatives, needs to be calculated in real time to fulfill an adaptive nonlinear dynamic inversion controller. Therefore, the directional derivatives of the TP simplex B-spline functions with regard to the control surface deflections are studied in this chapter. Five elements of the control allocation matrix $\mathbf{M}_{\text{CA}}$, see Eq. 8.2, are plotted in Figure 8.7. The actual values labeled with 'act' were obtained using the simulated wind tunnel data table from [114] and interpolation.



(a) Related to ailerons and elevators.

(b) Related to rudders.

Figure 8.7: Elements of the control allocation matrix $\mathbf{M}_{\text{CA}}$ with respect to aileron, elevator and rudder.

In Figure 8.7, $C_{l_{\delta_a}}$ stands for the directional derivative of C_l along the direction of δ_a. The $C_{l_{\delta_r}}$, $C_{n_{\delta_a}}$, $C_{n_{\delta_r}}$ and $C_{m_{\delta_e}}$ can be interpreted in the same way. In Figure 8.7(a), the estimated directional derivative is compared with the actual partial derivative. The actual value of $b \cdot C_{l_{\delta_a}}$ stays around -1.38 and reaches -0.78 at the 14.8^{th} second. The estimated value of $b \cdot C_{l_{\delta_a}}$ maintains the same sign, and is bounded between -1.42 and -0.85. The biggest difference between the actual value and the estimated value appears around the 15^{th} second, and the difference is around -0.65.

The estimated value of $b \cdot C_{l_{\delta_r}}$ is compared with its actual value in Figure 8.7(b). The estimated value stays close to 0.19 and maintains the same sign. The biggest difference is around 0.2, which shows between 15 and 19 seconds. As shown in Figure 8.7(b), the estimated $b \cdot C_{n_{\delta_a}}$ stays around -0.38, which is close to the actual value. Similarly, the estimated $b \cdot C_{n_{\delta_r}}$, as shown in Figure 8.7(b), keeps around -0.85. For both $b \cdot C_{n_{\delta_a}}$ and $b \cdot C_{n_{\delta_r}}$, the biggest difference between their estimated and actual values appears around the 15^{th} second, when the angle of attack has a big variation.

The estimated value of $\bar{c} \cdot C_{m_{\delta_e}}$ is plotted in Figure 8.7(a). Because the actual value can not be obtained directly by interpolation from the wind tunnel data table, the actual value shown in Figure 8.7(a) is the estimated value using small perturbation based centered difference. The estimated $\bar{c} \cdot C_{m_{\delta_e}}$ maintains between -2.65 and -0.8 with its sign negative definite. Except the period between the 12.5^{th} and 18^{th} seconds, the estimated value stays close to -1.65.

To summarize, from the adaptive NDI control point of view, the control effectiveness

matrix is not estimated optimally, which will therefore lead an non-optimal controller. Nevertheless, the estimated control effectiveness matrix should be applicable for the purpose of adaptive NDI control and does not destroy the stability of the NDI control system since its elements always stay very close to the actual values and are bounded in a proper region.

8.6.3. VALIDATION RESULTS USING A DATA SET FROM A CLOSED-LOOP FLIGHT TEST

In Section 8.6.2, a validation data set was obtained by running the open-loop F-16 aircraft with a 3211 type control inputs, and the function approximation power of the tensor-product simplex B-splines was demonstrated preliminarily. In this section, a typical flight task of the closed-loop F-16 aircraft system is considered to validate the TP simplex B-spline model structure further. A double-loop adaptive NDI angular controller is designed for the F-16 aircraft [114]. The controlled variables in the outer loop are ϕ, α and β. The controlled variables in the inner loop are the body angular rates p, q and r. For more details about how to design the adaptive NDI angular controller, the reader is referred to Lombaerts et al. [86][p.222-232]. A modified step-type reference command is given to the roll angle channel at the 3^{th} and 10^{th} seconds with a magnitude of 20 deg. Similarly, a modified step-type reference command is given to the angle of attack channel at the 15^{th} and 22^{th} seconds with a magnitude of 10 deg, see Appendix G.

The recorded control surface deflections for this angular command tracking task of the closed-loop F-16 control system are plotted in Figure 8.8. From Figure 8.8, it can be clearly seen that the ailerons and rudders are regulated by the controller in a coordinate way to track the step-type roll reference command at the 3^{th} and 10^{th} seconds. The commanded deflections of the elevator has a minimum value of -6.5 and a maximum value of -0.75. Due to the influence from the short period mode, the commanded elevator deflections show an oscillation after the 23^{th} second and have a trend of stabilizing the pitch motion.

The second data set, collected from the closed-loop system, is used as a validation data set aimed at showing the applicability of the tensor product splines in real-time application. The fitting performances in terms of the moment coefficients are plotted in Figure 8.9. Similar to the results shown in Figure 8.6, the estimated values of C_l and C_m shown in Figure 8.9 closely track the changes of their actual values, which are constructed from the wind tunnel data tables by interpolation.

The estimated values, reconstructed values and the actual values of C_n are plotted in Figure 8.9. Unlike the results shown in Figure 8.6, the estimated values of C_n show a minor steady estimation error between the 5^{th} and the 9^{th} seconds when compared with the actual values. This estimation difference is mainly due to the fact that the initial values of the B-coefficient vector are not obtained from a training data set with sufficient data coverage. Except the time period of 5-9 seconds, the fitting errors of C_n stay close to zero.

Similar to Section 8.6.2, the directional derivatives of the moment coefficients are calculated with regard to the aileron, elevator and rudder. The changes of the control derivatives are plotted in Figure 8.10. They are also compared with the actual

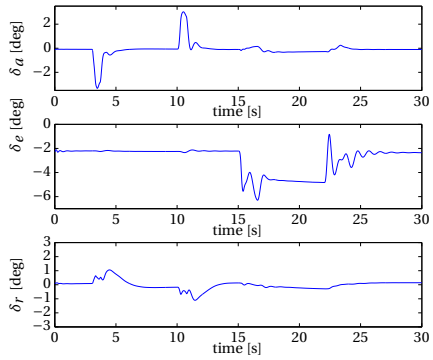


Figure 8.8: Recorded inputs of the closed-loop model, which is equipped with an ANDI controller.

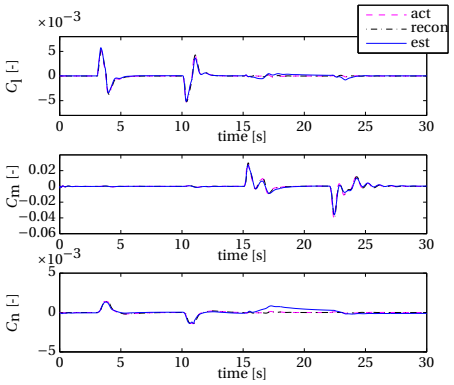
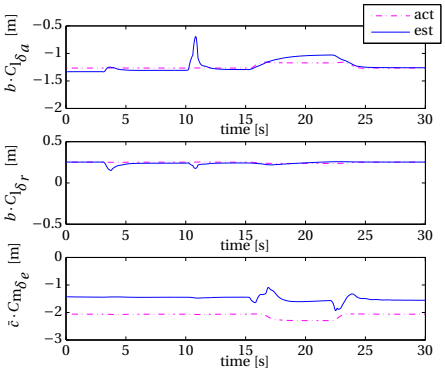
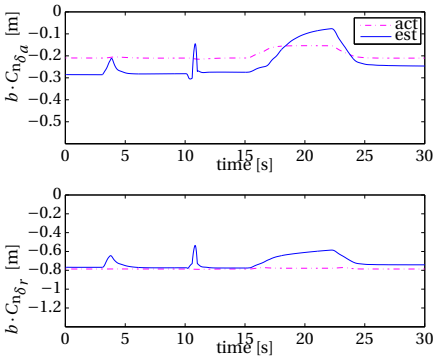


Figure 8.9: Comparison of the actual, reconstructed and estimated values of the non-dimensional moment coefficients, closed-loop data.

aerodynamic coefficients which are derived from the wind tunnel data tables presented in [114]. As shown in Figs. 8.10(a)-8.10(b), the estimated values of four elements of the control allocation matrix namely $b \cdot C_{l_{\delta_a}}$, $b \cdot C_{l_{\delta_r}}$,$b \cdot C_{n_{\delta_a}}$ and $b \cdot C_{n_{\delta_r}}$ are bounded and stay close to their actual values. The magnitude of the directional derivative of $\bar{c} \cdot C_m$ with respect to the elevator keeps higher between the 18^{th} and 23^{th} seconds than other period. This indicates that the elevator has a high efficiency when the angle of attack is large. For all of the elements in the control allocation matrix, the sign of their estimated values keeps unchanged during the whole simulation.



(a) Related to ailerons and elevators.



(b) Related to rudders.

Figure 8.10: Elements of the control allocation matrix $\mathbf{M}_{CA}$ with respect to aileron, elevator and rudder, closed-loop data.

To summarize, the modeling approach using the TP simplex B-splines is able to approximate the data set from a simulated closed-loop flight of an F-16. The estimated

control derivatives are properly bounded and stay close to the actual values. Comparing Figure 8.9 with Figure 8.6, it can also be seen that the magnitude of the moment coefficients of the second data set is much lower than that of the moment coefficients in the first data set. That is, the first data set stands for more agile maneuver than the second data set.

Remarks:

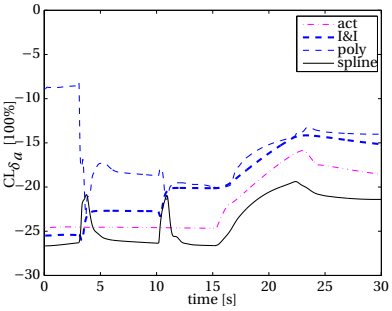
The directional derivatives of a highly nonlinear function which describes a dynamic system are always hard to estimate accurately or even not possible to estimate accurately under some circumstances. In the standard simplex B-spline functions, each argument is treated equally in the sense that each argument is assigned the same polynomial orders in the basis polynomials, see Eq. 4.4. Correspondingly, the standard simplex B-spline model structure has the chance to introduce some un-physical polynomial basis terms. These unnecessary terms are the terms that cause model mismatches.

In this chapter, only the output fitting errors are involved (evaluated) in the cost function. In this case, the first order directional derivatives in terms of control surface deflections cannot be guaranteed to track their corresponding actual values closely. However, if we have some *a priori* knowledge of the directional derivatives, e.g., the upper and lower bounds, these differential constraints can be taken into account during fitting the function outputs in order to enhance the estimation accuracy of the directional derivatives, see de Visser et al. [37].

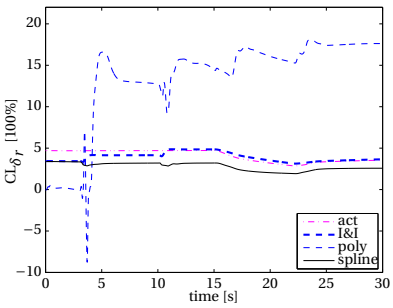
8.7. APPLICATION TO INCREMENTAL BACKSTEPPING AND THE COMPARISON

An incremental backstepping controller is designed for the F-16 aircraft model. The formulations of the aircraft motion equations used in this chapter are taken from van Oort et al. [163], and the incremental backstepping controller is designed using the method taken from Sonneveldt et al. [135] and Acquatella et al. [1]. The control allocation matrix demanded by the incremental controller is identified using the TP simplex B-splines, regular polynomial based method and Immersion & Invariance method, see [2], respectively. The I&I technique is initially aimed at making the tuning process of the modeling parameters independent from the tuning process of the controller parameters when using an integral (tuning function) adaptive control method. But it is used in this work to estimate a single effectiveness matrix with a focus on how well it can reflect the model parameter variations concerning the requirements from a controller. The identified control derivatives are compared to the true values obtained from the wind tunnel data tables. Thereafter, the flight performance of the incremental backstepping controller, which uses the control allocation matrix identified using the TP simplex B-spline identifier, is compared with the controllers using the control allocation matrix, which is identified using other identification algorithms.

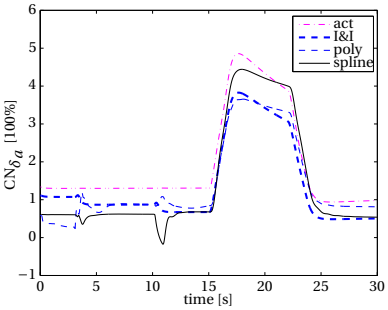
The changes in the control effectiveness parameters are plotted in Figure 8.11. As shown in the 5 figures, the directional derivatives calculated using the TP simplex B-splines are more precise than those obtained using either ordinary polynomials or the I&I method. The polynomial based method results in the worst estimation of the control effectiveness. The trends of the parameters are well captured by the TP simplex B-splines



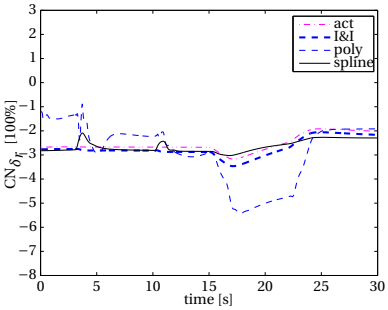
(a) $b \cdot C_{l_{\delta_a}}$.



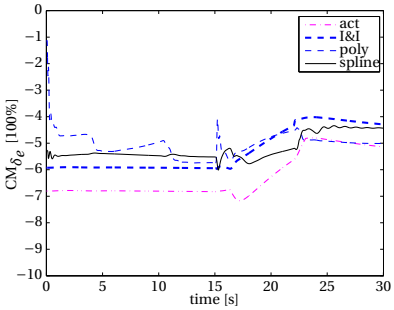
(b) $b \cdot C_{l_{\delta_r}}$.



(c) $b \cdot C_{n_{\delta_a}}$.



(d) $b \cdot C_{n_{\delta_r}}$.

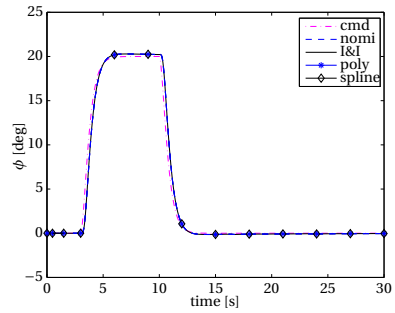


(e) $\bar{c} \cdot C_{m_{\delta_e}}$.

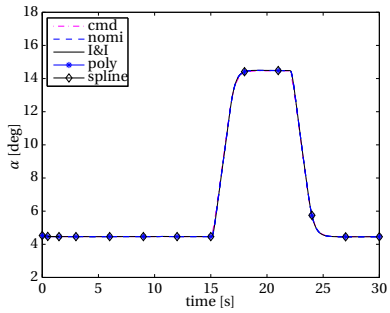
Figure 8.11: Elements of the control allocation matrix $\mathbf{M}_{\text{CA}}$, comparison between B-splines and the polynomials.

without evident oscillation in the estimating errors.

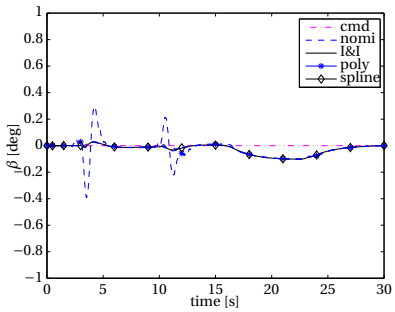
The angular and body angular rate control performances are depicted in Figs. 8.12(a)-8.12(f). The adaptive incremental backstepping controller using the TP simplex



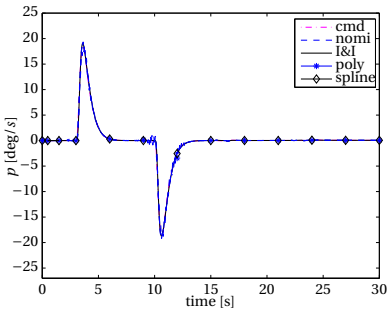
(a) Roll angle.



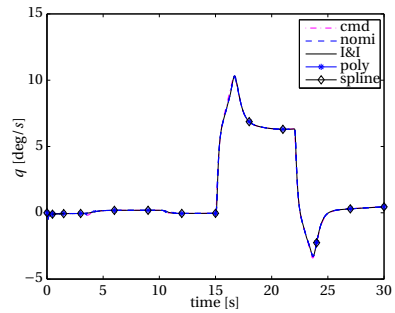
(b) Angle of attack.



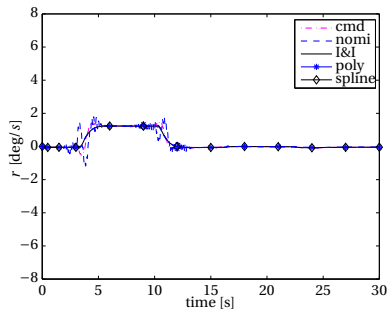
(c) Sideslip angle.



(d) Roll rate.



(e) Pitch rate.



(f) Yaw rate.

Figure 8.12: Control performance of an incremental backstepping controller with the control effectiveness matrix identified with polynomials and B-splines, respectively.

B-splines results in almost the same level of control performance as that of the other controllers, which use the I&I estimator or the actual effectiveness matrix, on the roll angle and the angle of attack. However, the flight controller using the polynomial based identifier results in worse control performance on the yaw rate r.

8.8. CONCLUSIONS

In this chapter, a novel compound structure for multivariate simplex B-splines is presented. The spline method using this compound model structure is referred to as the tensor-product (TP) simplex B-spline method. The TP simplex B-spline structure provides more options for model structure selection when using simplex B-splines. Different function inputs are allowed to be treated differently in terms of the selection of polynomial orders. Compared to the identification methods which use standard simplex B-splines, the linear regression modeling methods using TP simplex B-splines have a lower computational load requirement in the sense that it requires fewer B-coefficients. Specifically, if certain function inputs are less coupled with other inputs and the polynomial orders for those dimensions need not to be as high as other dimensions, the TP simplex B-splines, which require fewer B-coefficients than standard simplex B-splines, are recommended. In the TP simplex B-spline approach, the global basis vector maintains normalized, which contributes to the stability of the function approximation algorithm. When using TP simplex B-splines, the highest order of partial derivatives along each function input dimension is maintained.

The validation includes three parts. Firstly, the TP simplex B-spline based model identification method is validated using data sets generated using a F-16 nonlinear aircraft model. A 3-D and a 5-D function approximation problem related to the dimensionless pitch moment coefficient is used to demonstrate the advantages of TP simplex B-splines over standard simplex B-splines. Simulation results show that TP simplex B-splines have the potential to result in higher approximation power than standard simplex B-splines while using less B-coefficients. Secondly, TP simplex B-splines were further validated using simulated flight test data in a simulated flight test to show the applicability and efficiency of the new approach. The simulation results show that TP simplex B-splines are able to achieve an appropriate approximation accuracy in modeling all the three aerodynamic moment coefficients. In addition, the control derivatives can be estimated properly. Most precisely, the estimated control derivatives stay close to their actual values with their magnitudes bounded in a reasonable region. In the future, to estimate the directional derivatives more accurately, the differential constraints should be taken into account when identifying the B-coefficients. Thirdly, the control derivatives estimated using TP simplex B-splines are incorporated into an incremental backstepping (IBKS) controller designed for an F-16 aircraft model. Compared to an I&I approach, TP simplex B-splines can better capture the changing trend of the control effectiveness matrix. However, simulation results show that an IBKS controller is not sensitive to the estimation accuracy of the control effectiveness matrix.

9

CONCLUSIONS AND RECOMMENDATIONS

The goal of the research discussed in this thesis was twofold. First, to develop new powerful, cost-saving and computationally effective time-domain methods for identifying global models of nonlinear systems such as aircraft systems. Second, to extend acceleration measurements-based incremental control approaches to deal with structural or actuator failures occurring in an aircraft. The first was mainly developed for designing a model-based adaptive flight controller and for developing a model-based offline or online flight envelope protection approach. The second aimed at providing an alternative nonlinear reconfigurable control approach to fault-tolerant flight control. This goal is reflected by the main research question of this thesis:

Main research question
How can an advanced fault-tolerant flight control system be designed to increase the survivability of an aircraft.

Three new global model identification methods and two acceleration measurements-based incremental nonlinear flight control laws were developed to answer the above research question.

9.1. OVERVIEW OF THE WORK ON MODEL IDENTIFICATION

Among those three global model identification algorithms developed in Part One of this thesis, two of them use multivariate simplex B-splines (MVSB) and the third one was developed by combining a recursive kernel method with a support vector machine scheme. To enhance the computational efficiency of the parametric model identification methods using multivariate simplex B-spline polynomials, two recursive linear regression schemes were developed yielding two improved recursive linear-regression model

identification methods using MVSB. These two methods were validated using simulated flight test data generated with a high-fidelity nonlinear model of an F-16 aircraft. A novel recursive and parametric kernel method was developed for aerodynamic model identification. This focus of this work was to enhance the approximation power of a recursive and parametric kernel method by determining an optimal set of kernels for the kernel scheme. The adaptive kernel method was validated using a series of public available benchmark data sets well known to researchers from the field of pattern recognition [176, 179].

The global model identification methods function as follows. Firstly, a nominal aerodynamic model has to be trained *in advance* using pre-collected flight test data or wind-tunnel data. Then, the nominal model can be applied in a real-time situation, where the global model is allowed to be updated locally using flight test data for the current flight conditions. These global model identification methods then allow the estimated models to be stored for later re-use in a model-based adaptive flight controller for cases where the same flight conditions are revisited.

9.2. OVERVIEW OF ACCELERATION MEASUREMENTS-BASED INCREMENTAL CONTROL

A novel type of acceleration measurements-based incremental flight control laws was investigated and reported in Part Two of this thesis. The research aimed at providing a FTFC system with a powerful non-conventional flight control approach which could accommodate sudden structural or actuator failures occurring in an aircraft. In this part, a *sensor-based backstepping* approach is extended to handle sudden model changes in an aircraft caused by structural or actuator failures. A hybrid two-loop attitude controller and a joint two-loop angular controller were designed for the RECOVER model, the rate controller was designed using an acceleration measurements-based incremental controller. Both the hybrid two-loop angular controller and the joint two-loop angular controller were validated using the RECOVER model with a focus on dealing with two benchmark fault scenarios: namely a rudder runaway case and a flight 1862 engine separation scenario.

A global model identification method using tensor-product MVSB was developed. This new modeling method provides a user with more flexibility with respect to model structure selection and can enhance the clarity of a physical interpretation of the estimated model. Depending on a priori knowledge of a plant under estimation, the multivariate B-spline model can be chosen to have a different model structure, where a certain dimension of the spline function is treated differently from other dimensions. The tensor-product multivariate simplex B-splines method was applied to fit a nonlinear global model for a F-16 aircraft using simulated flight data generated using a high-fidelity nonlinear F-16 model. The control derivatives were derived from the tensor-product multivariate simplex B-spline model to provide a control effectiveness matrix for an incremental flight controller. How the estimation accuracy of the control effectiveness matrix, which is identified using the tensor-product simplex B-splines method, affects the performance of an incremental flight controller was investigated.

In the following, the conclusions obtained throughout the different chapters are

synthesized and presented. A general conclusion is given in Section 9.5. Finally, in Section 9.6, some recommendations for future research are presented.

9.3. RECURSIVE GLOBAL MODEL IDENTIFICATION METHODS

A global aerodynamic model is needed by a model-based fault-tolerant fight control (FTFC) system. Reconfigurable flight control methods based on a model-based adaptive control law have attracted a large amount of interest from the aerospace community over the past few decades because this type of methods extracts the most of the potential of the aircraft at current operating condition and can therefore enhance the aircraft's survivability in the case of failure. A full-envelope modular reconfigurable flight controller requires a global aerodynamic model which is valid in the entire flight envelope of an aircraft. At the meantime, the significance of flight envelope protection FEP has caught wide attention from engineers in the field of guidance navigation and control. One preferred branch of the FEP methods is the global aerodynamic model based offline or online FEP method. Central to the model-based FEP method is again an accurate global aerodynamic model. Once the aircraft states are given, an aerodynamic model identification problem is reduced into a system identification problem which can be solved using an *equation-error* approach. The *equation-error* approach is commonly based on a function approximation algorithm. A parametric function approximation scheme namely multivariate simplex B-splines MVSB and an adaptive kernel method are the potential candidates which can be well suited for the above-mentioned purpose. However, these two methods need to be improved either in computational efficiency or in approximation accuracy. The first subquestion related to the main research question was therefore:

The First Subquestion:

How can the candidate function approximation methods, i.e., **MVSB** and kernel methods, be improved in terms of their approximation accuracy and computational efficiency, to meet the needs of model-based adaptive control and online flight envelope protection **OFEP**?

To answer the first subquestion related to the main research question, the following were developed:

1. A substitution based solver was developed for solving an equality-constrained problem as in identifying a linear regression model using simplex B-splines (Chapter 3, Paper [138, 156]).

2. A recursive sequential routine was developed for updating a global linear regression model when using multivariate simplex B-splines (Chapter 4, Paper [141]).

3. A novel tensor-product (TP) multivariate simplex B-spline model structure was extended into a multidimensional case, and a detailed analysis of the properties of this model structure is given in Chapter 8 (Paper [142]).

4. A novel recursive parametric kernel method was developed by combining a regular recursive parametric kernel method with an improved recursive reduced least squares support vector regression (IRR-LSSVR) (Chapter 5, Paper [144]).

The linear regression model based on Bernstein basis polynomials was essential to all further applications of an aircraft model, for example model-based adaptive flight control and model-based flight envelope protection. However, the computational efficiency of existing parameter estimation techniques for deriving a multivariate simplex B-spline model still needs to be enhanced to enable the method to be applied in real-time. The two solutions to this end, proposed in this thesis, are summarized and concluded below.

Overview of Chapter 3 A substitution-based solver was developed for a linear regression modeling method using MVSB. The substitution solver was based on a singular value decomposition of the constraint matrix and a transformation using the vectors contained by the kernel space of the equality constraint matrix. The original equality constrained linear regression model identification problem was converted into a constraint-free linear regression modeling problem which could be solved using an ordinary least squares or recursive least squares identifier.

Conclusions of Chapter 3 The substitution-based mathematical transformation reduced the size of the original linear regression problem in the sense that the scale of the parameter covariance matrix was reduced, this resulted in an effective linear regression model updating algorithm when using MVSB. The computational complexity of the substitution based (SB) MVSB approach was given after an analysis of the complexity of the algorithm from the mathematical perspective. An equality constrained recursive least squares MVSB (ECRLS-MVSB) has a computational complexity of $\mathcal{O}\left(3m^2\right)$, and the SB-MVSB method has a computational complexity of $\mathcal{O}\left\{(m-r)\cdot(4m-3r)\right\}$ with m and r the total number of B-coefficients and the smoothness order, respectively. The analysis proved that the computational complexity of the SB-MVSB approach was much lower than that of an ECRLS-MVSB method, and the gap between these two methods was even larger when the rank of the constraint matrix was high. Thereafter, the SB-MVSB method was validated using simulated flight test data generated using a high-fidelity nonlinear F-16 model, and was compared with a model identification method using ordinary polynomial basis. These two methods were applied to model the nondimensional force and moment coefficients of an F-16 aircraft. The comparison of results showed that the SB-MVSB method led to higher approximation accuracy than the ordinary polynomial based method.

Main findings of Chapter 3:

A substitution based solver developed using a singular value decomposition can reduce the computational load of a recursive function approximation algorithm which uses a linear regression form of the **MVSB** model.

Overview of Chapter 4 A recursive sequential scheme was developed for deriving and updating a linear regression model when using MVSB. The aim was to enhance the

computational efficiency of the recursive MVSB method by transforming a global model identification problem at each time step into a per-simplex local-model-updating problem. This recursive model identification method is denoted as a recursive sequential (RS) MVSB approach in this thesis. In this method, the updating routine of the linear regression model consists of two consecutive steps. In step one of this approach, a per-simplex local model only is updated instead of updating the entire global model define on all the simplices. The high computational load resulted from the requirement to update the global covariance matrix is avoided. In step two, to allow for a smooth transition between different per-simplex local models, a *linear minimum mean square errors* (LMMSE) estimation, i.e., a linear projection with an optimality criterion, of the global B-coefficient vector is performed.

Conclusions of Chapter 4 The new recursive model identification method, i.e., RS-MVSB, was proven to be able to comply with the equality constraints resulting from the requirements on a smooth transition between different simplices. The computational complexity over time of the RS-MVSB and ECRLS-MVSB was given following an analysis from a mathematical operation prospective. The ECRLS-MVSB method has a computational complexity of $\mathcal{O}\left(3m^2\right)$ and the RS-MVSB method has a computational complexity of $\mathcal{O}\left(3\frac{m^2}{k^2}\right)$ with k the number of local models. The analytical results showed that the RS-MVSB was more computationally economic than the ECRLS-MVSB method. Two series of simulation experiments were performed. Firstly, the RS-MVSB method was validated using a 2-dimension data set and a 3-dimension data set designed artificially with white noise, and the method was compared with the ECRLS-MVSB method. Simulation results showed that RS-MVSB approach resulted in a much lower computational load than ECRLS-MVSB without evidently sacrificing its approximation power. As shown in the results for a 3-D demonstration example where the B-coefficient number was 2808, the RS-MVSB method was 60-70 times faster than the ECRLS-MVSB method when the codes were programmed in the Matlab running as interpreted scripts and run on a quad core PC (Intel Xeon E3-1270@3.40 GHZ, RAM 16.0 GB). Secondly, to demonstrate the high approximation power of the RS-MVSB method during aerodynamic model identification, the method was validated using a simulated flight test data generated using a high-fidelity nonlinear F-16 model, and it was compared with ECRLS-MVSB and a modeling method using an ordinary polynomial basis (OPB). The model structure selection of MVSB was also investigated with a focus on how the polynomial order and smoothness order affected the approximation power of the MVSB method. The simulation results showed that the new recursive global model identification method, i.e., RS-MVSB, led to a more accurate aerodynamic model than the ordinary polynomial based method and it yielded an accuracy comparative to that of ECRLS-MVSB as long as that the smoothness order was not chosen to be relatively high, e.g., larger than 2.

Main findings of Chapter 4:

An **RS-MVSB** method can reduce a computational load during model identification by only executing local updating of a local per-simplex model.

Overview of the algorithm part of Chapter 8 To provide more options for model structure selection when using MVSB, a novel tensor-product (TP) multivariate simplex B-spline (MVSB) model structure was studied. The tensor-product multivariate simplex B-splines (TP-MVSB) global model identification approach was extended from a single-dimension case into a multi-dimension case, and the properties of this new model structure were analyzed in detail. In a standard MVSB model, each dimension of inputs is treated identically where there is no chance to set different polynomial orders for different inputs. However, for some systems, a sufficiently accurate model can be achieved by choosing low polynomial orders for the spline polynomials in a certain dimension. For example, indicated in the *a priori* knowledge, a model under study might be affine in certain inputs. In the TP-MVSB approach, an input dimension, which is less correlated with other dimensions and does not require a polynomial order that is as high as other dimensions, can be treated differently from all the other inputs. The calculation procedure of the entire basis regression vector involving all inputs consists of two steps. First, a univariate spline basis vector is constructed for this chosen input, and a basis regression vector for the other remaining inputs is constructed using standard MVSB. Then, the entire basis regression vector is constructed by synthesizing those two basis vectors mentioned above using a tensor product operator. The TP-MVSB modeling structure was extended, i.e., generalized, to allow for dealing with the case where two or more dimensions are chosen to be treated specifically. In addition, the generic expression was given for calculating the partial derivatives with regard to any dimension of input.

Conclusions of the algorithm part of Chapter 8 Using a standard MVSB approach yields high-order basis polynomials, which may not be expected for certain inputs, during the construction of the basis regression vector. In contrast, these high-order basis polynomials can be set to be excluded by the basis regression vector when using a TP-MVSB approach. Compared to the standard MVSB, the TP-MVSB approach can reduce the scale of the MVSB model without sacrificing the approximation accuracy. As shown in one of the demonstration examples where the nondimensional pitch moment coefficient was estimated, the TP-MVSB approach required one third less B-coefficients than that needed by a standard MVSB method, but yielded a higher approximation accuracy. The TP-MVSB approach was applied to simulated flight test data sets generated using a high-fidelity nonlinear F-16 model. The simulation results showed that TP-MVSB can achieve a comparative approximation accuracy to that of standard MVSB when estimating the nondimensional moment coefficients of the F-16 aircraft.

Main findings of part one in Chapter 8:

The **TP-MVSB** modeling structure can treat different input dimensions differently, and it helps to reduce computational load by removing some unnecessary basis polynomials from the regression vector according to the *a priori* knowledge.

A consensus exists that kernel methods such as RBF neural networks have a high

approximation power. Considering this merit of the kernel methods, they were investigated in this thesis to perform aerodynamic model identification. Parametric type of kernel methods which uses a fixed number of kernels exists. One main challenge of using a parametric type recursive kernel method is how to determine an optimal subset of kernels in terms of kernel number, center position and bandwidth. This issue commonly has to be addressed by solving a global optimization problem, which is usually computationally time consuming.

Overview of Chapter 5 Two recursive model identification methods using kernels namely WV-LSSVR and GPK-LSSVR were proposed by combining a recursive parametric kernel method with an offline support vector machine namely improved recursive reduced least squares support vector regression (RR-LSSVR) (IRR-LSSVR). The IRR-LSSVR method, an offline model identification method, has been proven to be capable of choosing a lower number of more optimal support vectors than many other existing well-known support vector machines without sacrificing approximation accuracy. Therefore, IRR-LSSVR was applied in this work to determine number of kernels, center positions and bandwidths for the new recursive kernel methods, i.e., WV-LSSVR and GPK-LSSVR. This kernel determination process was done through an offline analysis of pre-collected modeling data sets. During a real-time application of the recursive kernel method, the linear regression model defined among the kernel space is updated using ordinary recursive least squares. The WV-LSSVR method uses ordinary Gaussian kernels, and GPK-LSSVR uses *Gaussian process kernels* which introduce a linear and a constant term in addition to an ordinary Gaussian kernel to enhance the local approximation power of the overall kernel method.

Conclusions of Chapter 5 During the determination of the support vectors from pre-collected data sets, the GPK-LSSVR and WV-LSSVR methods yielded the same computational complexity as the IRR-LSSVR method. During the recursive identification phase, the GPK-LSSVR method leads to a higher computational load than WV-LSSVR because the number of unknown parameters in the former is two times larger than that of the latter. The WV-LSSVR and GPK-LSSVR methods were validated using a set of 16 benchmark data sets, and they were compared with $k-means\,clustering$ radial basis function method (KMC-RBF) with a focus on approximation accuracy and computational efficiency. The comparison results showed that the GPK-LSSVR, WV-LSSVR and IRR-LSSVR methods always had much higher approximation powers than the KMC-RBF method with the change in the number of support vectors. Compared to WV-LSSVR, the GPK-LSSVR method leads to a slightly higher approximation power at the price of higher computational costs. Considering computational efficiency, although the GPK-LSSVR method takes into account different data trends among different subdomains using an extended kernel, the GPK-LSSVR method is not always to be preferred. The choice between the WV-LSSVR and GPK-LSSVR methods should be performed based on the characteristics of the model, i.e., the nonlinearity level of the system.

Main findings of Chapter 5:

The **IRR-LSSVR** is an efficient algorithm that is capable of providing optimal kernel parameters for an ordinary recursive parametric kernel method, and local kernel extension helps to capture the local dynamics of a model.

9.4. ACCELERATION MEASUREMENTS-BASED RECONFIGURABLE CONTROL

Model-based adaptive flight control has advantages, for example the useful system dynamics which contribute to the system stability can be selected not to be counteracted when designing a controller. However, an online model derived for an adaptive controller might not be sufficiently accurate during high maneuvering flight or structural aircraft failure [88], which will make the control performance of the flight controller deteriorate or even make the controller unstable. As an alternative, acceleration measurements-based flight control law, which does not require real-time full aerodynamic model information, and is therefore easier to certify, was investigated for the research reported in this thesis. The second subquestion related to the main research question was:

The Second Subquestion:

What are the benefits of using an acceleration measurements-based control approach, i.e., the sensor-based backstepping, as an alternative to a model-based adaptive control approach, when designing a reconfigurable flight controller to deal with aircraft failures in a generic **FTFC** system?

In order to answer the second subquestion related to the main research question, the following were developed.

1. A hybrid two-loop attitude (angular) controller was designed with the angular rate controller designed using sensor-based backstepping (Chapter 6, Paper [139]).

2. A joint two-loop angular controller was designed with the angular rate controller designed using sensor-based backstepping control law (Chapter 7, Papers [140, 143]).

3. The tensor-product multivariate simplex B-spline model structure was employed to calculate the control derivatives to provide control effectiveness matrix for acceleration measurements-based incremental body angular rate controller (Chapter 8, Paper [142]).

An incremental type control approach, namely approximate dynamic inversion based on singular perturbation theory and Tikhonov's theorem, has been developed

by Hovakimyan et al. [59] for a non-affine in control nonlinear system. The stability of the closed-loop system, the controller of which is designed based on a Lyapunov stability function, has been proved under realistic assumptions. In 2011, Falkena et al. [45] reformulated this *incremental* control approach to allow for direct utilization of the measurements of angular accelerometers. A two-loop attitude (angular) controller has been designed for a Diamond-42 small fixed wing aircraft using the incremental control law indicated as *sensor based backstepping (SBB)*. However, this new control approach has not yet been applied to a large civil transportation aircraft for the purpose of fault tolerant control. In the work reported in this thesis, the SBB control approach was extended to deal with benchmark structural failures occurring in a Boeing 747 aircraft. Two two-loop angular controllers, namely a hybrid NDI/SBB controller and a joint SBB controller, were developed for the RECOVER model. The two-loop hybrid NDI/SBB controller was developed by combining an NDI based outer controller with an inner controller designed using the SBB control law. The two-loop SBB angular controller is an improved version of the two-loop hybrid NDI/SBB angular controller in the sense that a Lyapunov function based multi-loop controller design technique, i.e., recursive backstepping, was used. Both angular controllers were validated within a four-loop autopilot using the RECOVER model. The four-loop autopilot consists of an altitude control loop and a flight path angle control loop designed using proportional-integral-derivative in addition to the aforementioned two-loop angular and body angular rate control loop.

Overview of Chapter 6 The hybrid controller indicated as hybrid NDI/SBB angular controller consists of two control loops namely an angular loop and a body angular rate loop. The inner body angular rate controller was designed using the singular perturbation theory based sensor based backstepping (SBB) control approach, and the outer attitude controller was designed using the NDI control law. The commanded control inputs by the outer angular loop were taken as reference commands for the inner angular rate loop. In the controller design, the control allocation problem was simplified by bounding a number of the control surfaces into a group. The controller was applied to the RECOVER model of a Boeing 747 aircraft, and evaluated using rudder runaway and EL AL flight 1862 benchmark fault scenarios developed by the GARTEUR FM-AG 16 group. The differential thrust control was introduced to replace the rudder deflection control to counteract the undesirable yawing moment induced by the stuck rudder.

Conclusions of Chapter 6 Compared with the classic model-based adaptive nonlinear dynamic inversion (ANDI) control approach or model-based adaptive backstepping control law, the hybrid NDI/SBB angular control setup needs less online model information. The numerical simulation results showed that the proposed hybrid NDI/SBB angular controller can preserve the safety of the aircraft even when the aforementioned failures occur, and can ensure a zero tracking error performance for the roll angle and the pitch angle commands as long as the aircraft is still controllable with the remaining valid control surfaces.

Main findings of Chapter 6:

A hybrid NDI/SBB two-loop controller has the capability to accommodate sudden model changes due to the mechanism of incremental approximation control in the body angular rate control loop.

Overview of Chapter 7 A two-loop joint SBB angular controller was developed for the RECOVER model, the heart of which is a Boeing 747 aircraft model. Unlike the hybrid NDI/SBB angular controller, those two control loops in the joint SBB angular controller were designed cooperatively using a recursive backstepping technique starting from the angular loop. At each backstepping step, a Lyapunov's stability function was used in designing a controller to stabilize the system. Similar to other incremental type nonlinear flight controllers, measurements of the body angular accelerations were required. In the research reported in this thesis, the angular accelerations were numerically calculated from the filtered body angular rates using a differentiator. However, in a real application, the body angular accelerations can be directly obtained from the angular accelerometers instead, which is currently under investigation by many research groups.

Conclusions of Chapter 7 The numerical simulation results showed that the double-loop joint SBB angular controller can lead to zero tracking errors as long as the given angular reference commands are within the safe flight envelope. That is, the new joint SBB angular controller was shown to be able to stabilize asymptotically the angular reference tracking system under both benchmark faults under consideration. Compared with the hybrid SBB angular controller, the new joint SBB method leads to better zero-hold performance in controlling sideslip angle when an aircraft is flying in the nominal condition or under the engine separation scenario. Under the rudder runaway fault scenario, the new controller presented in this thesis leads to equivalent control performance of sideslip to that of the hybrid SBB angular controller mentioned above.

Main findings of Chapter 7:

A joint SBB two-loop angular controller results in better or at least equivalent control performance compared to a hybrid NDI/SBB controller because a recursive backstepping design strategy based on Lyapunov function is introduced.

Overview of the control-related part of Chapter 8 How the control effectiveness matrix influences the performance of the acceleration measurements-based incremental flight controller was investigated. The TP-MVSB approach presented in Chapter 8 and the immersion and invariance (I&I) estimator were used to estimate the effectiveness matrix of the F-16 aircraft. Although the I&I technique initially is not aiming for high accuracy but rather for improving system stability, it was used in this work to provide a reasonable, consistent estimation of a single control effectiveness matrix with a focus on reflecting the variations of the model parameters concerning the requirements for a controller. The

control effectiveness matrices identified using both methods were applied to a regular incremental backstepping controller to show how the estimation accuracy of the control derivatives affect the performance of the incremental flight controller.

Conclusions of the control-related part of Chapter 8 The TP-MVSB method was able to provide a reasonably accurate estimation of a control effectiveness matrix for a nonlinear incremental backstepping controller. In comparison, TP-MVSB leads to a better changing rate estimation than the I&I estimator when estimating the effectiveness matrix. Simulation results showed that a slight difference on the control effectiveness matrix made no difference to the attitude flight control performance as long as the conducted flight task required moderate rather than high maneuvering. That is, for an aircraft like F-16, the performance of an incremental controller is not susceptible to the estimation accuracy of the control effectiveness matrix.

Main findings of part two in Chapter 8:

The **TP-MVSB** method leads to a reasonably accurate estimation of the control effectiveness matrix, which can meet the need of an incremental backstepping flight controller.

9.5. GENERAL CONCLUSIONS

Synthesizing the results obtained and given throughout the individual chapters, the following general conclusions can be drawn.

Stability issue related to model-based adaptive control:
As is well known, a model-based, i.e., modular, adaptive control approach is hard to certify concerning system stability because most of the modular approaches can only guarantee *input-to-state* stability. The main reason why modular adaptive control laws cannot guarantee closed-loop system stability is that a large amount of modular adaptive control laws suffer from the weakness of the *certainty equivalence* principle. To achieve closed-loop stability, the identified model in a modular adaptive control system is required to be sufficiently accurate. The recursive identification methods using multivariate simplex B-splines, reported in this thesis, can achieve highly accurate aerodynamic model, and the model outputs are always bounded by the maximum B-coefficient. These properties of the simplex B-splines make the aerodynamic model easier to certify compared to other more complex modeling methods, e.g. radial basis function neural networks.

Both model-based and acceleration measurements-based flight control methods have advantages and drawbacks, the users should select the methods according to the specific situation they are facing, e.g. concerning physical limitations of the onboard computers or the availability of angular accelerometers.

Modular adaptive control *VS* Angular Acceleration Measurements-based Control:

The joint sensor-based backstepping controller based on the singular perturbation theory, presented in Chapter 7, is recommended if the measurements or an accurate estimation of angular acceleration are obtainable, because this incremental type control law has a low computational-load requirement for the onboard computer. A model-based adaptive reconfigurable flight control law becomes the preferred option if the onboard computer has adequate computational power, or, an accurate aerodynamic model is also required for other components in the entire flight control system, e.g., fault detection and diagnosis unit and flight envelope protection unit. Model-based control has the advantage of being able to design a flight controller which maintains useful damping terms in the closed-loop system.

9.6. RECOMMENDATIONS

The work presented in this thesis gives rise to new questions and research directions, some recommendations for further studies are presented below.

Three global model identification methods, namely SB-MVSB, RS-MVSB and GPK-LSSVR, were validated using modeling data. The development of these three global model identification methods is a good start towards model-based adaptive flight control and flight envelope protection. However, these methods should be incorporated into a model-based adaptive flight control law or an online flight envelope protection scheme to demonstrate further the benefits of using a real-time accurate global aerodynamic model.

The number of simplices in a triangulation increases dramatically with the increase in the pre-determined vertices in each dimension and input dimensions when using a simplex B-spline model identification method. To enhance the computational efficiency, especially when the inputs are of high dimension, efficient optimization algorithms concerning the data coverage for each simplex should be developed aimed at constructing a triangulation with less simplices.

The directional derivatives of a function, e.g., the control derivatives, are hard to estimate accurately or even not possible to estimate accurately under some circumstances. In this thesis, only the output fitting errors are involved (evaluated) in the cost function. In this case, the first order directional derivatives in terms of control surface deflections cannot be guaranteed to track their corresponding actual values closely. However, if we have some *a priori* knowledge of the directional derivatives, e.g., the upper and lower bounds, these differential constraints can be taken into account during fitting the function outputs to enhance the estimation accuracy of the directional derivatives, see de Visser et al. [37].

No fault detection & isolation (FDI) algorithm was studied for the work reported in this thesis. When implementing the fault-tolerant flight controller in Chapters 6 and 7, the assumption was made that the type of failures and its timing were known. Fault detection of structural failures, actuator failures and sensor failures need to be investigated.

How the sensor noise of angular accelerometers affects the control performance of an acceleration measurements-based incremental controller should be further investigated before the sensor based backstepping control approach can be applied in real flight. In addition, the influence of time delays occurring in actuators or engine systems should be further investigated, especially during controller designs for real aircraft. Further research into acceleration measurements based reconfigurable control should include tests on the SIMONA simulator, realistic test-flights with UAVs and possibly the research aircraft of TU Delft.

The sensor-based backstepping (SBB) controller should be tested on the SIMONA simulator to get more feedback on its use from experienced pilots. Comments from pilots can be used to help the controller designers to choose better controller gains or time-scale parameters for the SBB controller.

Taking into account the estimation of a time-varying wind using a Kalman filter, how the wind dynamics should be described or modeled in the time-derivative equations of the aircraft kinematics should be investigated. For example, a turbulence model such as the NASA Dryden model should be investigated with a focus on its suitability for the aforementioned purpose.

With respect to online flight envelope protection (OFEP), constructing an offline global aerodynamic model for each aircraft fault scenario should be investigated. A regular online global model identification method can only update the model locally given a limited number of incoming data points, therefore, the identified aerodynamic model is more likely not valid for the entire flight envelope in a relatively short period after any sudden structural or actuator failures happen. Nevertheless, OFEP requires a global-valid aerodynamic model immediately after a failure occurs to an aircraft to estimate the current safe flight envelope. This shortcoming of online global model identification methods could possibly be circumvented by constructing an offline global aerodynamic model for each fault scenario.

To avoid the *curse-of-dimensionality* problem associated with OFEP when using a reachability analysis approach, i.e., evolution of the Hamilton-Jacobi PDEs, more efficient mathematical tools such as the *max-plus* method [95], which is *curse-of-dimensionality-free*, should be investigated.

A

KINEMATIC EQUATIONS FOR KALMAN FILTERS

The kinematic equations of a fixed-wing aircraft and the corresponding augmented expressions are given for applications of an extended Kalman filter in this section.

A.1. THE KINEMATIC EQUATIONS

The general set of nonlinear system equations describing the kinematics of the aircraft is given as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}_m(t), \Theta, t] + \mathbf{G}[\mathbf{x}]\mathbf{w}(t),\ \mathbf{x}(t_0) = \mathbf{x}_0 \tag{A.1}$$

$$\mathbf{z}_m(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}_m(t), \Theta, t] + \mathbf{v}(t),\ t = t_i,\ i = 1, 2, \ldots \tag{A.2}$$

where Eq. A.1 is known as the kinematic state equation with input noise vector $\mathbf{w}$ and expression Eq. B.2 is called the observation equation with output noise vector $\mathbf{v}$. The nonlinear vector functions $\mathbf{f}$ and $\mathbf{h}$ may depend both implicitly (via $\mathbf{x}$ and $\mathbf{u}_m$) and explicitly on t and it will be assumed that both $\mathbf{f}$ and $\mathbf{h}$ are continuous and continuously differentiable with respect to all elements of $\mathbf{x}$ and $\mathbf{u}_m$. The system equation variables are defined below.

$$\mathbf{x} = \left[x_{\text{GS}}, y_{\text{GS}}, z_{\text{GS}}, u_{\text{AS}}, v_{\text{AS}}, w_{\text{AS}}, \phi, \theta, \psi\right]^\top \tag{A.3}$$

$$\mathbf{u}_m = \mathbf{u} + \lambda + \mathbf{w} = \left[A_x, A_y, A_z, p, q, r\right]^\top + \left[\lambda_x, \lambda_y, \lambda_z, \lambda_p, \lambda_q, \lambda_r\right]^\top + \mathbf{w} \tag{A.4}$$

$$\Theta = \lambda = \left[\lambda_x, \lambda_y, \lambda_z, \lambda_p, \lambda_q, \lambda_r\right]^\top \tag{A.5}$$

$$\mathbf{z}_m = \left[x_{\text{GPS}}, y_{\text{GPS}}, z_{\text{GPS}}, u_{\text{GPS}}, v_{\text{GPS}}, w_{\text{GPS}}, \phi_{\text{GPS}}, \theta_{\text{GPS}}, \psi_{\text{GPS}}, V_{\text{TAS}}, \alpha_{\text{ADS}}, \beta_{\text{ADS}}\right]^\top \tag{A.6}$$

with the subscripts GS, AS, GPS and ADS the abbreviations of 'ground speed', 'air speed', 'global positioning system' and 'air data system'. Because the measurements from an inertial navigation system contain biases, 9 variables in $\mathbf{z}_m$ applied here should be the

measurements taken from a GPS system. Specifically, three antennas, i.e., receiver for GPS signals, are needed in order to obtain ϕ_{GPS}, θ_{GPS}, and ψ_{GPS}.

The full expression of the kinematic equations, see Eq. A.1, can be written as follows, where the aircraft is considered as a rigid body above a flat non-rotating earth:

$$\dot{x}_{\text{GS}} = \left[u_{\text{AS}}\cos\theta + \left(v_{\text{AS}}\sin\phi + w_{\text{AS}}\cos\phi\right)\sin\theta\right]\cos\psi - \left(v_{\text{AS}}\cos\phi - w_{\text{AS}}\sin\phi\right)\sin\psi + U_{\text{wind}} \tag{A.7}$$

$$\dot{y}_{\text{GS}} = \left[u_{\text{AS}}\cos\theta + \left(v_{\text{AS}}\sin\phi + w_{\text{AS}}\cos\phi\right)\sin\theta\right]\sin\psi + \left(v_{\text{gb}}\cos\phi - w_{\text{gb}}\sin\phi\right)\cos\psi + V_{\text{wind}} \tag{A.8}$$

$$\dot{z}_{\text{GS}} = -u_{\text{AS}}\sin\theta + \left(v_{\text{AS}}\sin\phi + w_{\text{AS}}\cos\phi\right)\cos\theta + W_{\text{wind}} \tag{A.9}$$

$$\dot{u}_{\text{AS}} = A_x - g\sin\theta + r\,v_{\text{AS}} - q\,w_{\text{AS}} \tag{A.10}$$

$$\dot{v}_{\text{AS}} = A_y - g\cos\theta\sin\phi + p\,w_{\text{AS}} - r\,u_{\text{AS}} \tag{A.11}$$

$$\dot{w}_{\text{AS}} = A_z - g\cos\theta\cos\phi + q\,u_{\text{AS}} - p\,v_{\text{AS}} \tag{A.12}$$

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{A.13}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{A.14}$$

$$\dot{\psi} = q\frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta} \tag{A.15}$$

$$\tag{A.16}$$

A.2. AUGMENTED KINEMATIC EQUATIONS FOR EXTENDED KALMAN FILTER

During the application of the EKF, the aircraft state vector $\mathbf{x}$ from Eq. A.3 should be augmented into $\mathbf{x}_{\text{aug}} = [\mathbf{x}, \Theta]$. The augmented state vector $\mathbf{x}_{\text{aug}}$ contains 18 augmented states.

$$\mathbf{x}_{\text{aug}} = \left[x_{\text{GS}}, y_{\text{GS}}, z_{\text{GS}}, u_{\text{AS}}, v_{\text{AS}}, w_{\text{AS}}, \phi, \theta, \psi, \lambda_x, \lambda_y, \lambda_z, \lambda_p, \lambda_q, \lambda_r, U_{\text{wind}}, V_{\text{wind}}, W_{\text{wind}}\right] \tag{A.17}$$

with $U_{\text{wind}}, V_{\text{wind}}, W_{\text{wind}}$ the wind speed with regard to the ground, and $u_{\text{AS}}, v_{\text{AS}}, w_{\text{AS}}$ the airspeed body components. For simplicity reason, $\mathbf{x}_{\text{aug}}$ will be written as $\mathbf{x}$ in the remainder of this chapter.

The set of continuous state equations $\mathbf{f}[\mathbf{x}(t), \mathbf{u}_m(t), \theta]$ are given as follows:

$$\mathbf{f}[\mathbf{x}(t), \mathbf{u}_m(t), \theta] = \begin{bmatrix} \left[u_{\text{AS}} \cos\theta + \left(v_{\text{AS}} \sin\phi + w_{\text{AS}} \cos\phi\right) \sin\theta\right] \cos\psi - \left(v_{\text{AS}} \cos\phi - w_{\text{AS}} \sin\phi\right) \sin\psi + U_{\text{wind}} \\ \left[u_{\text{AS}} \cos\theta + \left(v_{\text{AS}} \sin\phi + w_{\text{AS}} \cos\phi\right) \sin\theta\right] \sin\psi + \left(v_{\text{AS}} \cos\phi - w_{\text{AS}} \sin\phi\right) \cos\psi + V_{\text{wind}} \\ -u_{\text{AS}} \sin\theta + \left(v_{\text{AS}} \sin\phi + w_{\text{AS}} \cos\phi\right) \cos\theta + W_{\text{wind}} \\ \left(A_{x_m} - \lambda_x\right) - g \sin\theta + \left(r_m - \lambda_r\right) v_{\text{AS}} - \left(q_m - \lambda_q\right) w_{\text{AS}} \\ \left(A_{y_m} - \lambda_y\right) - g \cos\theta \sin\phi + \left(p_m - \lambda_p\right) w_{\text{AS}} - \left(r_m - \lambda_r\right) u_{\text{AS}} \\ \left(A_{z_m} - \lambda_z\right) - g \cos\theta \cos\phi + \left(q_m - \lambda_q\right) u_{\text{AS}} - \left(p_m - \lambda_p\right) v_{\text{AS}} \\ \left(p_m - \lambda_p\right) + \left(q_m - \lambda_q\right) \sin\phi \tan\theta + \left(r_m - \lambda_r\right) \cos\phi \tan\theta \\ \left(q_m - \lambda_q\right) \cos\phi - \left(r_m - \lambda_r\right) \sin\phi \\ \left(q_m - \lambda_q\right) \frac{\sin\phi}{\cos\theta} + \left(r_m - \lambda_r\right) \frac{\cos\phi}{\cos\theta} \\ \mathbf{0}_{9\times 1} \end{bmatrix} \tag{A.18}$$

The linear operation matrix $\mathbf{G}[\mathbf{x}(t)]$ becomes:

$$\mathbf{G}[\mathbf{x}(t)] = \begin{bmatrix} & & & & \mathbf{0}_{3\times 6} & \\ -1 & 0 & 0 & 0 & w_{\text{AS}} & -v_{\text{AS}} \\ 0 & -1 & 0 & -w_{\text{AS}} & 0 & u_{\text{AS}} \\ 0 & 0 & -1 & v_{\text{AS}} & -u_{\text{AS}} & 0 \\ 0 & 0 & 0 & -1 & -\sin\phi \tan\theta & -\cos\phi \tan\theta \\ 0 & 0 & 0 & 0 & -\cos\phi & \sin\phi \\ 0 & 0 & 0 & 0 & -\frac{\sin\phi}{\cos\theta} & -\frac{\cos\phi}{\cos\theta} \\ & & & & \mathbf{0}_{9\times 6} & \end{bmatrix} \tag{A.19}$$

The observation equation set becomes:

$$
\begin{aligned}
x_{\text{GPS}} &= x_{\text{GS}} + \nu_x \\
y_{\text{GPS}} &= y_{\text{GS}} + \nu_y \\
z_{\text{GPS}} &= z_{\text{GS}} + \nu_z \\
u_{\text{GPS}} &= \left[u_{\text{AS}}\cos\theta + \left(v_{\text{AS}}\sin\phi + w_{\text{AS}}\cos\phi\right)\sin\theta\right]\cos\psi \\
&\quad - \left(v_{\text{AS}}\cos\phi - w_{\text{AS}}\sin\phi\right)\sin\psi + U_{\text{wind}} + \nu_u \\
v_{\text{GPS}} &= \left[u_{\text{AS}}\cos\theta + \left(v_{\text{AS}}\sin\phi + w_{\text{AS}}\cos\phi\right)\sin\theta\right]\sin\psi \\
&\quad + \left(v_{\text{AS}}\cos\phi - w_{\text{AS}}\sin\phi\right)\cos\psi + V_{\text{wind}} + \nu_v \\
w_{\text{GPS}} &= -u_{\text{AS}}\sin\theta + \left(v_{\text{AS}}\sin\phi + w_{\text{AS}}\cos\phi\right)\cos\theta + W_{\text{wind}} + \nu_w \\
\phi_{\text{GPS}} &= \phi + \nu_\phi \\
\theta_{\text{GPS}} &= \theta + \nu_\theta \\
\psi_{\text{GPS}} &= \psi + \nu_\psi \\
V_{\text{TAS}} &= \sqrt{u_{\text{AS}}^2 + v_{\text{AS}}^2 + w_{\text{AS}}^2} + \nu_V \\
\alpha_{\text{ADS}} &= \arctan\left(\frac{w_{\text{AS}}}{u_{\text{AS}}}\right) + \nu_\alpha \\
\beta_{\text{ADS}} &= \arctan\left(\frac{v_{\text{AS}}}{\sqrt{u_{\text{AS}}^2 + w_{\text{AS}}^2}}\right) + \nu_\beta
\end{aligned}
\tag{A.20}
$$

B

New Formulation of Kinematic Equations for Applying an Extended Kalman Filter

The classic formulation of the kinematic equations for applying an extended Kalman filter, i.e. the kinematic equations shown in Appendix A, was derived under the assumption that the wind speed was time-invariant. This work is aimed at generalizing the formulation of a set of kinematic equations for applying an extended Kalman filter (EKF) and attention is paid to estimate non-zero mean time-varying wind speeds. Due to this reason, the body components of the ground velocity presents in the differential equations of linear velocities.

B.1. Nonlinear aircraft kinematics model

The general set of nonlinear system equations describing the kinematics of the aircraft is given as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}_m(t), \Theta, t] + \mathbf{G}[\mathbf{x}]\mathbf{w}(t), \ \mathbf{x}(t_0) = \mathbf{x}_0 \tag{B.1}$$

$$\mathbf{z}_m(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}_m(t), \Theta, t] + \mathbf{v}(t), \ t = t_i, i = 1, 2, \ldots \tag{B.2}$$

where Eq. B.1 is known as the kinematic state equation with input noise vector $\mathbf{w}$ and expression Eq. B.2 is called the observation equation with output noise vector $\mathbf{v}$. The nonlinear vector functions $\mathbf{f}$ and $\mathbf{h}$ may depend both implicitly (via $\mathbf{x}$ and $\mathbf{u}_m$) and explicitly on t and it will be assumed that both $\mathbf{f}$ and $\mathbf{h}$ are continuous and continuously differentiable with respect to all elements of $\mathbf{x}$ and $\mathbf{u}_m$. The system equation variables

are defined below.

$$\mathbf{x} = \left[x_{\text{GS}}, y_{\text{GS}}, z_{\text{GS}}, u_{\text{gb}}, v_{\text{gb}}, w_{\text{gb}}, \phi, \theta, \psi\right]^\top \tag{B.3}$$

$$\mathbf{u}_m = \mathbf{u} + \lambda + \mathbf{w} = \left[A_x, A_y, A_z, p, q, r\right]^\top + \left[\lambda_x, \lambda_y, \lambda_z, \lambda_p, \lambda_q, \lambda_r\right]^\top + \mathbf{w} \tag{B.4}$$

$$\Theta = \lambda = \left[\lambda_x, \lambda_y, \lambda_z, \lambda_p, \lambda_q, \lambda_r\right]^\top \tag{B.5}$$

$$\mathbf{z}_m = \left[x_{\text{GPS}}, y_{\text{GPS}}, z_{\text{GPS}}, u_{\text{GPS}}, v_{\text{GPS}}, w_{\text{GPS}}, \phi_{\text{GPS}}, \theta_{\text{GPS}}, \psi_{\text{GPS}}, V_{\text{TAS}}, \alpha_{\text{ADS}}, \beta_{\text{ADS}}\right]^\top \tag{B.6}$$

with the subscripts GS, AS, GPS and ADS the abbreviations of 'ground speed', 'air speed', 'global positioning system' and 'air data system'. Because the measurements from an inertial navigation system contain biases, 9 variables in $\mathbf{z}_m$ are the measurements taken from a GPS system. Specifically, three antenas, i.e., receiver for GPS signals, are needed in order to obtain ϕ_{GPS}, θ_{GPS}, ψ_{GPS}.

The full expression of the **nonlinear** kinematic equations, see Eq. B.1, can be written as follows, where the aircraft is condidered as a rigid body above a flat non-rotating earth:

$$\dot{x}_{\text{GS}} = \left[u_{\text{gb}}\cos\theta + \left(v_{\text{gb}}\sin\phi + w_{\text{gb}}\cos\phi\right)\sin\theta\right]\cos\psi - \left(v_{\text{gb}}\cos\phi - w_{\text{gb}}\sin\phi\right)\sin\psi \tag{B.7}$$

$$\dot{y}_{\text{GS}} = \left[u_{\text{gb}}\cos\theta + \left(v_{\text{gb}}\sin\phi + w_{\text{gb}}\cos\phi\right)\sin\theta\right]\sin\psi + \left(v_{\text{gb}}\cos\phi - w_{\text{gb}}\sin\phi\right)\cos\psi \tag{B.8}$$

$$\dot{z}_{\text{GS}} = -u_{\text{gb}}\sin\theta + \left(v_{\text{gb}}\sin\phi + w_{\text{gb}}\cos\phi\right)\cos\theta \tag{B.9}$$

$$\dot{u}_{\text{gb}} = A_x - g\sin\theta + r\,v_{\text{gb}} - q\,w_{\text{gb}} \tag{B.10}$$

$$\dot{v}_{\text{gb}} = A_y - g\cos\theta\sin\phi + p\,w_{\text{gb}} - r\,u_{\text{gb}} \tag{B.11}$$

$$\dot{w}_{\text{gb}} = A_z - g\cos\theta\cos\phi + q\,u_{\text{gb}} - p\,v_{\text{gb}} \tag{B.12}$$

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{B.13}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{B.14}$$

$$\dot{\psi} = q\frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta} \tag{B.15}$$

$$\tag{B.16}$$

B.2. AUGMENTED STATE EQUATIONS AND OBSERVATION EQUATIONS

During the application of the EKF, the aircraft state vector $\mathbf{x}$ from Eq. B.3 should be augmented into $\mathbf{x}_{\text{aug}} = [\mathbf{x}, \Theta]$. The augmented state vector $\mathbf{x}$ contains 18 augmented states.

$$\mathbf{x}_{\text{aug}} = \left[x_{\text{GS}}, y_{\text{GS}}, z_{\text{GS}}, u_{\text{gb}}, v_{\text{gb}}, w_{\text{gb}}, \phi, \theta, \psi, \lambda_x, \lambda_y, \lambda_z, \lambda_p, \lambda_q, \lambda_r, U_{\text{wind}}, V_{\text{wind}}, W_{\text{wind}}\right] \tag{B.17}$$

with $U_{\text{wind}}, V_{\text{wind}}, W_{\text{wind}}$ the wind speed with regard to the ground, and $u_{\text{gb}}, v_{\text{gb}}, w_{\text{gb}}$ the ground velocity body components. For simlicity reason, $\mathbf{x}_{\text{aug}}$ will be written as $\mathbf{x}$ in the remaindar of this chapter.

The state equations are similar to those included in chapter 4 of the dissertation of Lombaerts except that the wind components should be removed. Similarly, the wind components should be also removed from Eqs.(4.40-4.42). On the contrary, during the

calculations of the true air speed V_{TAS}, α and β, we have to correct the ground velocity body components with wind or turbulence velocity components (converted to body) when wind and turbulence present.

The transformation matrix from the earth to the body reference frame:

$$T_{o\rightarrow b} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \begin{pmatrix}\sin\phi\sin\theta\cos\psi \\ -\cos\phi\sin\psi\end{pmatrix} & \begin{pmatrix}\sin\phi\sin\theta\sin\psi \\ +\cos\phi\cos\psi\end{pmatrix} & \sin\phi\cos\theta \\ \begin{pmatrix}\cos\phi\sin\theta\cos\psi \\ +\sin\phi\sin\psi\end{pmatrix} & \begin{pmatrix}\cos\phi\sin\theta\sin\psi \\ -\sin\phi\cos\psi\end{pmatrix} & \cos\phi\cos\theta \end{bmatrix} \tag{B.18}$$

The ground wind speeds are converted into the body wind speeds:

$$\begin{bmatrix} u_{\text{wind}} \\ v_{\text{wind}} \\ w_{\text{wind}} \end{bmatrix} = T_{o\rightarrow b} \begin{bmatrix} U_{\text{wind}} \\ V_{\text{wind}} \\ W_{\text{wind}} \end{bmatrix} \tag{B.19}$$

The output V_{TAS} is:

$$V_{\text{TAS}} = \sqrt{\left(u_{\text{gb}} - u_{\text{wind}}\right)^2 + \left(v_{\text{gb}} - v_{\text{wind}}\right)^2 + \left(w_{\text{gb}} - w_{\text{wind}}\right)^2} \tag{B.20}$$

The output α becomes:

$$\alpha = \arctan\left(\frac{w_{\text{gb}} - w_{\text{wind}}}{u_{\text{gb}} - u_{\text{wind}}}\right) \tag{B.21}$$

The output β is:

$$\beta = \arctan\left(\frac{v_{\text{gb}} - v_{\text{wind}}}{\sqrt{\left(u_{\text{gb}} - u_{\text{wind}}\right)^2 + \left(w_{\text{gb}} - w_{\text{wind}}\right)^2}}\right) \tag{B.22}$$

To summarize, the set of continuous state equations $\mathbf{f}[\mathbf{x}(t), \mathbf{u}_m(t), \theta]$ are given as follows:

$$\mathbf{f}[\mathbf{x}(t), \mathbf{u}_m(t), \theta] = \begin{bmatrix} \left[u_{\text{gb}}\cos\theta + \left(v_{\text{gb}}\sin\phi + w_{\text{gb}}\cos\phi\right)\sin\theta\right]\cos\psi - \left(v_{\text{gb}}\cos\phi - w_{\text{gb}}\sin\phi\right)\sin\psi \\ \left[u_{\text{gb}}\cos\theta + \left(v_{\text{gb}}\sin\phi + w_{\text{gb}}\cos\phi\right)\sin\theta\right]\sin\psi + \left(v_{\text{gb}}\cos\phi - w_{\text{gb}}\sin\phi\right)\cos\psi \\ -u_{\text{gb}}\sin\theta + \left(v_{\text{gb}}\sin\phi + w_{\text{gb}}\cos\phi\right)\cos\theta \\ \left(A_{x_m} - \lambda_x\right) - g\sin\theta + \left(r_m - \lambda_r\right)v_{\text{gb}} - \left(q_m - \lambda_q\right)w_{\text{gb}} \\ \left(A_{y_m} - \lambda_y\right) - g\cos\theta\sin\phi + \left(p_m - \lambda_p\right)w_{\text{gb}} - \left(r_m - \lambda_r\right)u_{\text{gb}} \\ \left(A_{z_m} - \lambda_z\right) - g\cos\theta\cos\phi + \left(q_m - \lambda_q\right)u_{\text{gb}} - \left(p_m - \lambda_p\right)v_{\text{gb}} \\ \left(p_m - \lambda_p\right) + \left(q_m - \lambda_q\right)\sin\phi\tan\theta + \left(r_m - \lambda_r\right)\cos\phi\tan\theta \\ \left(q_m - \lambda_q\right)\cos\phi - \left(r_m - \lambda_r\right)\sin\phi \\ \left(q_m - \lambda_q\right)\frac{\sin\phi}{\cos\theta} + \left(r_m - \lambda_r\right)\frac{\cos\phi}{\cos\theta} \\ \mathbf{0}_{9\times 1} \end{bmatrix} \tag{B.23}$$

The linear operation matrix $\mathbf{G}[\mathbf{x}(t)]$ becomes:

B

$$\mathbf{G}[\mathbf{x}(t)] = \begin{bmatrix} & & & & \mathbf{0}_{3\times 6} & \\ -1 & 0 & 0 & 0 & w_{gb} & -v_{gb} \\ 0 & -1 & 0 & -w_{gb} & 0 & u_{gb} \\ 0 & 0 & -1 & v_{gb} & -u_{gb} & 0 \\ 0 & 0 & 0 & -1 & -\sin\phi\tan\theta & -\cos\phi\tan\theta \\ 0 & 0 & 0 & 0 & -\cos\phi & \sin\phi \\ 0 & 0 & 0 & 0 & -\frac{\sin\phi}{\cos\theta} & -\frac{\cos\phi}{\cos\theta} \\ & & & & \mathbf{0}_{9\times 6} & \end{bmatrix} \tag{B.24}$$

The observation equation set becomes:

$$x_{GPS} = x_{GS} + v_x \tag{B.25}$$

$$y_{GPS} = y_{GS} + v_y \tag{B.26}$$

$$z_{GPS} = z_{GS} + v_z \tag{B.27}$$

$$u_{GPS} = \left[u_{gb}\cos\theta + \left(v_{gb}\sin\phi + w_{gb}\cos\phi\right)\sin\theta\right]\cos\psi - \left(v_{gb}\cos\phi - w_{gb}\sin\phi\right)\sin\psi + v_u \tag{B.28}$$

$$v_{GPS} = \left[u_{gb}\cos\theta + \left(v_{gb}\sin\phi + w_{gb}\cos\phi\right)\sin\theta\right]\sin\psi + \left(v_{gb}\cos\phi - w_{gb}\sin\phi\right)\cos\psi + v_v \tag{B.29}$$

$$w_{GPS} = -u_{gb}\sin\theta + \left(v_{gb}\sin\phi + w_{gb}\cos\phi\right)\cos\theta + v_w \tag{B.30}$$

$$\phi_{GPS} = \phi + v_\phi \tag{B.31}$$

$$\theta_{GPS} = \theta + v_\theta \tag{B.32}$$

$$\psi_{GPS} = \psi + v_\psi \tag{B.33}$$

$$V_{TAS} = \sqrt{\left(u_{gb} - u_{wind}\right)^2 + \left(v_{gb} - v_{wind}\right)^2 + \left(w_{gb} - w_{wind}\right)^2} + v_V \tag{B.34}$$

$$\alpha_{ADS} = \arctan\left(\frac{w_{gb} - w_{wind}}{u_{gb} - u_{wind}}\right) + v_\alpha \tag{B.35}$$

$$\beta_{ADS} = \arctan\left(\frac{v_{gb} - v_{wind}}{\sqrt{\left(u_{gb} - u_{wind}\right)^2 + \left(w_{gb} - w_{wind}\right)^2}}\right) + v_\beta \tag{B.36}$$

with u_{wind}, v_{wind} and w_{wind} calculated from Eq.Eq. B.19.

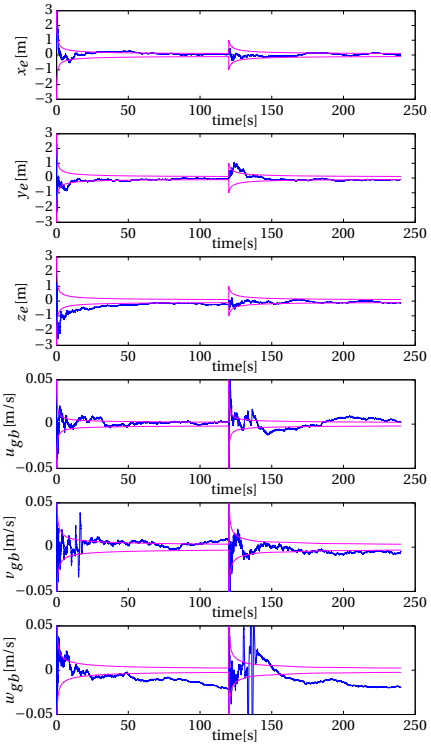
Remarks:

During the application of the EKF, the Jacobian matrix of V_{TAS}, α, β with respect to $[u_{gb}, v_{gb}, w_{gb}, \phi, \theta, \psi, U_{wind}, V_{wind}, W_{wind}]$ is a 3 by 9 matrix, which can be easily calculated using the Maple software. The matlab implementation codes are automatically generated by the Maple software.

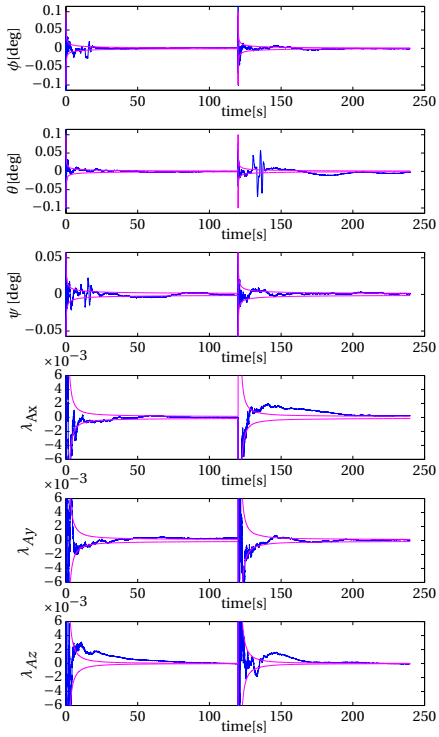
B.3. VALIDATION RESULTS USING THE FLIGHT TEST DATA OF A CESSNA CITATION II

The two 'de3211' and 'dr3211' flight test data sets (i.e. each contains 12001 data points, see the lecture notes of the course AE 4-394) of the Cessna Citation II laboratory fixed-

wing aircraft are used to validate the newly formulated kinematic equations for applying an extended Kalman Filter. The validation results are plotted in Figs.B.1(a)-B.2(b).



(a) State fitting errors, part one.

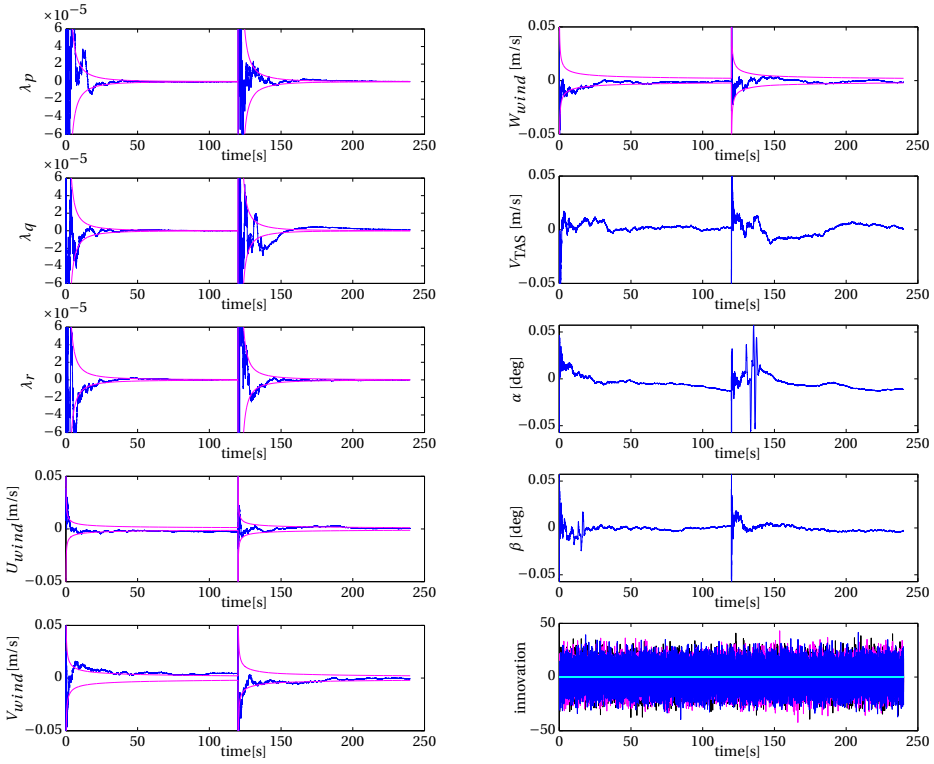


(b) State fitting errors, part two.

Figure B.1: Aircraft state estimation results for states, sensor biases.

As shown in Figs. B.1-B.2, the fitting errors of all the states are almost bounded by a 1-σ boundary which is calculated from the diagnal elements of the covariance matrix. All the innovations related to the observations are located in a proper range. Simulation results shown in Figs. B.1-B.2 have validated the correctness of the new kinematic equations formulated for apply an extended Kalman filter. A conclusion can be drawn that an extended Kalman filter based on the newly formulated kinematic expressions is able to estimate non-zero mean time-invariant wind speeds.

B



(a) State fitting errors, part three.

(b) State fitting errors, part four, and innovations.

Figure B.2: Aircraft state estimation results for sensor biases, wind speeds, and reconstructed air data measurements.

B.4. CONCLUSION

How aircraft states can be estimated when an aircraft is encountering a non-zero mean time-varying wind, i.e. a turbulence, was investigated, and some preliminary results were given. The classic formulation of the kinematic equations for an extended Kalman filter (EKF) was derived under the assumption that the wind speed was time-invariant. To deal with a non-zero mean time-varying wind, the description equations of the kinematic equations were generalized for the application of an EKF by replacing the chosen state variables by new variables defined in a different reference coordinate system. Simulated flight test data generated using a Cessna Citation II model with a *time-invariant* non-zero mean wind were used to validate the correctness of the new formulation for using an EKF. In the designed demonstration exapble, only time-invariant wind speed estimation was accounted for.

The simulation results showed that an EKF using the new kinematic model description can achieve an unbiased and reasonable, accurate state estimation. However, how the wind dynamics should be modeled, e.g. using the NASA Dryden wind turbulence model, in the augmented expression of the kinematic equations still needs to be investigated before the EKF can be really applied to deal with non-zero mean time-varying wind.

C

A Specific Aerodynamic Model Structure for RECOVER Model using Polynomials

For later use, the configurations of an Boeing 747-100/200 aircraft are given in Figure C.1.

When using a polynomial-based aerodynamic model identification method presented in [88, 89] in real-time applications, a predefined model structure for a Boeing 747 aircraft is chosen as follows. For more details, the reader is referred to [88].

$$\begin{aligned} C_{\mathrm{X}} = & C_{\mathrm{X}_0} + C_{\mathrm{X}_\alpha}\alpha + C_{\mathrm{X}_{\alpha^2}}\alpha^2 + C_{\mathrm{X}_q}\frac{q\bar{c}}{V} + C_{\mathrm{X}_{\delta_{e_{ir}}}}\left|\delta_{e_{ir}}\right| + C_{\mathrm{X}_{\delta_{e_{il}}}}\left|\delta_{e_{il}}\right| + C_{\mathrm{X}_{\delta_{eor}}}\left|\delta_{e_{or}}\right| \\ & + C_{\mathrm{X}_{\delta_{e_{ol}}}}\left|\delta_{e_{ol}}\right| + C_{\mathrm{X}_{i_h}}\left|X_{i_h}\right| + C_{\mathrm{X}_{\delta_{sp_1}}}\delta_{sp_1} + \ldots + C_{\mathrm{X}_{\delta_{sp_{12}}}}\delta_{sp_{12}} + C_{\mathrm{X}_{\delta_{fo}}}\delta_{fo} \\ & + C_{\mathrm{X}_{\delta_{fi}}}\delta_{fi} + C_{\mathrm{X}_{EPR_1}}EPR_1 + \ldots + C_{\mathrm{X}_{EPR_4}}EPR_4 + C_{\mathrm{X}_\beta}\beta + C_{\mathrm{X}_p}\frac{pb}{2V} + C_{\mathrm{X}_r}\frac{rb}{2V} \end{aligned} \tag{C.1}$$

$$\begin{aligned} C_{\mathrm{Z}} = & C_{\mathrm{Z}_0} + C_{\mathrm{Z}_\alpha}\alpha + C_{\mathrm{Z}_q}\frac{q\bar{c}}{V} + C_{\mathrm{Z}_{\delta_{e_{ir}}}}\delta_{e_{ir}} + C_{\mathrm{Z}_{\delta_{e_{il}}}}\delta_{e_{il}} + C_{\mathrm{Z}_{\delta_{eor}}}\delta_{e_{or}} + C_{\mathrm{Z}_{\delta_{e_{ol}}}}\delta_{e_{ol}} \\ & + C_{\mathrm{Z}_{i_h}}Z_{i_h} + C_{\mathrm{Z}_{\delta_{sp_1}}}\delta_{sp_1} + \ldots + C_{\mathrm{Z}_{\delta_{sp_{12}}}}\delta_{sp_{12}} + C_{\mathrm{Z}_{\delta_{fo}}}\delta_{fo} + C_{\mathrm{Z}_{\delta_{fi}}}\delta_{fi} \\ & + C_{\mathrm{Z}_{EPR_1}}EPR_1 + \ldots + C_{\mathrm{Z}_{EPR_4}}EPR_4 + C_{\mathrm{Z}_\beta}\beta + C_{\mathrm{Z}_p}\frac{pb}{2V} + C_{\mathrm{Z}_r}\frac{rb}{2V} \end{aligned} \tag{C.2}$$

$$\begin{aligned} C_{\mathrm{m}} = & C_{\mathrm{m}_0} + C_{\mathrm{m}_\alpha}\alpha + C_{\mathrm{m}_q}\frac{q\bar{c}}{V} + C_{\mathrm{m}_{\delta_{e_{ir}}}}\delta_{e_{ir}} + C_{\mathrm{m}_{\delta_{e_{il}}}}\delta_{e_{il}} + C_{\mathrm{m}_{\delta_{eor}}}\delta_{e_{or}} + C_{\mathrm{m}_{\delta_{e_{ol}}}}\delta_{e_{ol}} \\ & + C_{\mathrm{m}_{i_h}}Z_{i_h} + C_{\mathrm{m}_{\delta_{sp_1}}}\delta_{sp_1} + \ldots + C_{\mathrm{m}_{\delta_{sp_{12}}}}\delta_{sp_{12}} + C_{\mathrm{m}_{\delta_{fo}}}\delta_{fo} + C_{\mathrm{m}_{\delta_{fi}}}\delta_{fi} \\ & + C_{\mathrm{m}_{EPR_1}}EPR_1 + \ldots + C_{\mathrm{m}_{EPR_4}}EPR_4 + C_{\mathrm{m}_\beta}\beta + C_{\mathrm{m}_p}\frac{pb}{2V} + C_{\mathrm{m}_r}\frac{rb}{2V} \end{aligned} \tag{C.3}$$

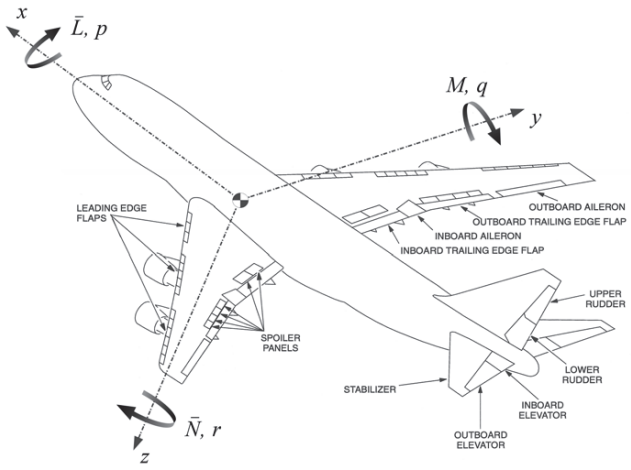


Figure C.1: Boeing 747-100/200 control surface arrangements, body axes and moment definitions, from [132].

$$\begin{aligned} C_{\mathrm{Y}} = & C_{\mathrm{Y}_0} + C_{\mathrm{Y}_\beta}\beta + C_{\mathrm{Y}_p}\frac{pb}{2V} + C_{\mathrm{Y}_r}\frac{rb}{2V} + C_{\mathrm{Y}_{\delta a_{ir}}}\delta a_{ir} + C_{\mathrm{Y}_{\delta a_{il}}}\delta a_{il} + C_{\mathrm{Y}_{\delta a_{or}}}\delta a_{or} \\ & + C_{\mathrm{Y}_{\delta a_{ol}}}\delta a_{ol} + C_{\mathrm{Y}_{\delta ru}}\delta ru + C_{\mathrm{Y}_{\delta rl}}\delta rl + C_{\mathrm{Y}_{\delta sp_1}}\delta sp_1 + \ldots + C_{\mathrm{Y}_{\delta sp_{12}}}\delta sp_{12} \\ & + C_{\mathrm{Y}_\alpha}\alpha + C_{\mathrm{Y}_q}\frac{q\bar{c}}{V} + C_{\mathrm{Y}_{EPR_1}}EPR_1 + \ldots + C_{\mathrm{Y}_{EPR_4}}EPR_4 \end{aligned} \tag{C.4}$$

$$\begin{aligned} C_{\mathrm{l}} = & C_{\mathrm{L}_0} + C_{\mathrm{l}_\beta}\beta + C_{\mathrm{l}_p}\frac{pb}{2V} + C_{\mathrm{l}_r}\frac{rb}{2V} + C_{\mathrm{l}_{\delta a_{ir}}}\delta a_{ir} + C_{\mathrm{l}_{\delta a_{il}}}\delta a_{il} + C_{\mathrm{l}_{\delta a_{or}}}\delta a_{or} \\ & + C_{\mathrm{l}_{\delta a_{ol}}}\delta a_{ol} + C_{\mathrm{l}_{\delta ru}}\delta ru + C_{\mathrm{l}_{\delta rl}}\delta rl + C_{\mathrm{l}_{\delta sp_1}}\delta sp_1 + \ldots + C_{\mathrm{l}_{\delta sp_{12}}}\delta sp_{12} \\ & + C_{\mathrm{l}_\alpha}\alpha + C_{\mathrm{l}_q}\frac{q\bar{c}}{V} + C_{\mathrm{l}_{EPR_1}}EPR_1 + \ldots + C_{\mathrm{l}_{EPR_4}}EPR_4 \end{aligned} \tag{C.5}$$

$$\begin{aligned} C_{\mathrm{n}} = & C_{\mathrm{n}_0} + C_{\mathrm{n}_\beta}\beta + C_{\mathrm{n}_p}\frac{pb}{2V} + C_{\mathrm{n}_r}\frac{rb}{2V} + C_{\mathrm{n}_{\delta a_{ir}}}\delta a_{ir} + C_{\mathrm{n}_{\delta a_{il}}}\delta a_{il} + C_{\mathrm{n}_{\delta a_{or}}}\delta a_{or} \\ & + C_{\mathrm{n}_{\delta a_{ol}}}\delta a_{ol} + C_{\mathrm{n}_{\delta ru}}\delta ru + C_{\mathrm{n}_{\delta rl}}\delta rl + C_{\mathrm{n}_{\delta sp_1}}\delta sp_1 + \ldots + C_{\mathrm{n}_{\delta sp_{12}}}\delta sp_{12} \\ & + C_{\mathrm{n}_\alpha}\alpha + C_{\mathrm{n}_q}\frac{q\bar{c}}{V} + C_{\mathrm{n}_{EPR_1}}EPR_1 + \ldots + C_{\mathrm{n}_{EPR_4}}EPR_4 \end{aligned} \tag{C.6}$$

D

PROPERTIES OF RECURSIVE SEQUENTIAL MVSB AND PROOFS

Three theorems about the convergence and smoothness maintaining property of the new algorithm will be formulated and proved. In this section, $\mathbf{c}$ is the discontinuous global B-coefficient vector derived from Eq. 4.16 independently and $\tilde{\mathbf{c}}$ is the parameter vector derived by recursive sequential method as shown in Eq. 4.26.

Theorem 1: Convergence

Given the discontinuous global B-coefficient vector $\mathbf{c} = \left[c_1^\top, c_2^\top, ..., c_i^\top, ..., c_n^\top\right]^\top$ with c_i denoting the parameter vector of the i^{th} module, there exists an unique converged value $\tilde{\mathbf{c}}_f$ for the sequential estimation method that makes $\lim_{k\to+\infty} \tilde{\mathbf{c}} = \tilde{\mathbf{c}}_f$ with k the number of training data points.

Proof:

Given $\mathbf{c} = \left[c_1^\top, c_2^\top, ..., c_i^\top, ..., c_n^\top\right]^\top$ with n the number of modules, a unique c_{if} exists for $\lim_{k\to+\infty} c_i = c_{if}$ where c_i denotes the local estimation vector, and it contains the estimated parameters of module i. We can get a unique $\mathbf{c}_f = \left[c_{1f}^\top, c_{2f}^\top, ..., c_{if}^\top, ..., c_{nf}^\top\right]^\top$ satisfying $\lim_{k\to+\infty} \mathbf{c} = \mathbf{c}_f$. According to Eq. 4.26, the smooth global B-coefficient vector $\tilde{\mathbf{c}}$ is the least-mean-square-error (LMSE) estimation of $\mathbf{c}$, and this process is a linear operator. Therefore, we can derive a unique $\tilde{\mathbf{c}}_f = f_{LMSE}\left(\mathbf{c}_f\right)$ that makes $\lim_{k\to+\infty} \tilde{\mathbf{c}} = \tilde{\mathbf{c}}_f$. □

Remarks:

The first recursive step of this sequential estimation method deals with the identification problem on a small module. Nearly all the existing least squares based online estimation methods can cope with it and assure that c_i converges to c_{if}.

Theorem 2: Smoothness

The global continuity of the spline function derived from the recursive sequential identification method is maintained after each smoothing step.

Proof:

In the calculation of the smooth global B-coefficient vector $\tilde{\mathbf{c}} \in \mathbb{R}^{m\times 1}$, Eq. 4.26 holds

for every time step. Apparently, $\widetilde{\mathbf{c}}$ is a linear combination of orthonormal vector bases $\{\eta_1, \eta_2, ...\eta_{n-1}, \eta_n\}$. Because $\eta_1, \eta_2, ..., \eta_{n-1}, \eta_n$ are columns of U_2 as shown in Eq. 4.25, they constitute bases of the kernel space [76] for the equality constraints matrix $\mathbf{H}$. Therefore, the constraint equations $\mathbf{H} \cdot \widetilde{\mathbf{c}} = 0$ are fully satisfied at each time step. Because of this, the global continuity character of the global spline function is maintained through the smooth global B-coefficient vector $\widetilde{\mathbf{c}}$. □

In the conventional method, the constrained least squares problem is transformed into an unconstrained least squares problem at the first step by introducing Lagrangian multipliers, resulting in a globally optimal solution in the least squares sense [37]. In contrast, the kernel space projection based recursive sequential identification method proposed in this paper can only achieve optimality during each single stage. This property will be elaborated in the following theorem.

Theorem 3: Reduced approximation power

Given $\hat{\mathbf{c}}$ the B-coefficients estimated with the batch least squares method from [37], and $\widetilde{\mathbf{c}}$ the B-coefficients estimated using RS-LS, it holds that $\left|\left|\mathbf{y} - \mathbf{X} \cdot \widetilde{\mathbf{c}}\right|\right| \geq \left|\left|\mathbf{y} - \mathbf{X} \cdot \hat{\mathbf{c}}\right|\right|$.

Proof:

The proof of this theorem is based on the loss of orthogonality of $\widetilde{\epsilon} = \mathbf{y} - \mathbf{X} \cdot \widetilde{\mathbf{c}}$ to the hyperplane spanned by col $\mathbf{X}$ after the smoothing step of RS-LS. In the first step of RS-LS, the rough B-coefficient vector $\mathbf{c}$ is estimated such $\epsilon = \mathbf{y} - \mathbf{X} \cdot \mathbf{c}$ is orthogonal to col $\mathbf{X}$, i.e. $\epsilon \in \perp \text{col } \mathbf{X}$. During the smoothing step, the rough global B-coefficient vector $\mathbf{c}$ is projected into a linear subspace determined by the columns of $\mathbf{U}_2$ from Eq. 4.18. In general, we have col $\mathbf{U}_2 \notin \{\text{col } \mathbf{X}, \perp \text{col } \mathbf{X}\}$ and therefore $\widetilde{\epsilon} \notin \perp \text{col } \mathbf{X}$, which proves the theorem. □

E

Calculate Sideslip Angle Related NDI Term

From the flight dynamics, the sideslip angle is defined as:

$$\beta = \arcsin\left(\frac{v}{V}\right) \tag{E.1}$$

with

$$V = \sqrt{u^2 + v^2 + w^2} \tag{E.2}$$

By taking the time derivatives of β, Eq. E.1 becomes:

$$\begin{aligned}\dot{\beta} &= \frac{\dot{v}V - v\dot{V}}{V\sqrt{V^2 - v^2}} = \frac{\dot{v}}{\sqrt{V^2 - v^2}} - \frac{\dot{v}V}{V\sqrt{V^2 - v^2}} \\ &= \frac{\dot{v}}{\sqrt{u^2 + w^2}} - \frac{v\left(u\dot{u} + v\dot{v} + w\dot{w}\right)}{\left(u^2 + v^2 + w^2\right)\sqrt{u^2 + w^2}}\end{aligned} \tag{E.3}$$

From the flight dynamics, the following equations hold:

$$\begin{bmatrix}\dot{u} \\ \dot{v} \\ \dot{w}\end{bmatrix} = \frac{1}{m}\begin{bmatrix}X \\ Y \\ Z\end{bmatrix} - \begin{bmatrix}qw - rv \\ ru - pw \\ pv - qu\end{bmatrix} + g\begin{bmatrix}\sin\theta \\ \sin\phi\sin\theta \\ \cos\phi\cos\theta\end{bmatrix} \tag{E.4}$$

Substituting Eq. E.4 into the first term of Eq. E.3 results in:

$$\begin{aligned}\frac{\dot{v}}{\sqrt{u^2 + w^2}} &= \frac{1}{\sqrt{u^2 + w^2}}\left(\frac{Y}{m} - ru + pw + g\sin\phi\cos\theta\right) \\ &= \frac{1}{\sqrt{u^2 + w^2}}\left(\frac{Y}{m} + g\sin\phi\cos\theta\right) + \begin{bmatrix}\frac{w}{\sqrt{u^2 + w^2}} & 0 & \frac{-u}{\sqrt{u^2 + w^2}}\end{bmatrix}\begin{bmatrix}p \\ q \\ r\end{bmatrix}\end{aligned} \tag{E.5}$$

Using similar substitution, the second term of Eq. E.3 becomes:

$$\begin{aligned}\frac{v(u\dot{u}+v\dot{v}+w\dot{w})}{V^2\sqrt{u^2+w^2}} = \frac{v}{V^2\sqrt{u^2+w^2}}\Bigg\{u\left[\frac{X}{m}-(qw-rv)-g\sin\theta\right] \\ +v\left[\frac{Y}{m}-(ru-pw)+g\sin\phi\cos\theta\right]+w\left[\frac{Z}{m}-(pv-qu)+g\cos\phi\cos\theta\right]\Bigg\}\end{aligned} \tag{E.6}$$

Eq. E.6 can be further simplified as:

$$\begin{aligned}\frac{v(u\dot{u}+v\dot{v}+w\dot{w})}{V^2\sqrt{u^2+w^2}} = & \frac{v}{V^2\sqrt{u^2+w^2}}\left[u\left(\frac{X}{m}-g\sin\theta\right)+v\left(\frac{Y}{m}+g\sin\phi\cos\theta\right)+w\left(\frac{Z}{m}+g\cos\phi\cos\theta\right)\right. \\ & \left.-(qw-rv)-(ru-pw)-(pv-qu)\right] \\ = & \frac{v}{V^2\sqrt{u^2+w^2}}\left[u\left(\frac{X}{m}-g\sin\theta\right)+v\left(\frac{Y}{m}+g\sin\phi\cos\theta\right)+w\left(\frac{Z}{m}+g\cos\phi\cos\theta\right)\right]\end{aligned} \tag{E.7}$$

Combining Eq. E.5 with Eq. E.7 yields:

$$\begin{aligned}\dot{\beta} = & \frac{1}{\sqrt{u^2+w^2}}\left(\frac{Y}{m}+g\sin\phi\cos\theta\right)+\begin{bmatrix}\frac{w}{\sqrt{u^2+w^2}} & 0 & \frac{-u}{\sqrt{u^2+w^2}}\end{bmatrix}\begin{bmatrix}p\\q\\r\end{bmatrix} \\ & -\frac{v}{V^2\sqrt{u^2+w^2}}\left[u\left(\frac{X}{m}-g\sin\theta\right)+v\left(\frac{Y}{m}+g\sin\phi\cos\theta\right)+w\left(\frac{Z}{m}+g\cos\phi\cos\theta\right)\right] \\ = & \frac{1}{\sqrt{u^2+w^2}}\left[\frac{-uv}{V^2}\left(\frac{X}{m}-g\sin\theta\right)+\left(1-\frac{v}{V^2}\right)\left(\frac{Y}{m}+g\sin\phi\cos\theta\right)-\frac{vw}{V^2}\left(\frac{Z}{m}+g\cos\phi\cos\theta\right)\right] \\ & +\begin{bmatrix}\frac{w}{\sqrt{u^2+w^2}} & 0 & \frac{-u}{\sqrt{u^2+w^2}}\end{bmatrix}\begin{bmatrix}p\\q\\r\end{bmatrix} \\ = & \frac{1}{\sqrt{u^2+w^2}}\left[\frac{-uv}{V^2}\left(A_x-g\sin\theta\right)+\left(1-\frac{v}{V^2}\right)\left(A_y+g\sin\phi\cos\theta\right)-\frac{vw}{V^2}\left(A_z+g\cos\phi\cos\theta\right)\right] \\ & +\begin{bmatrix}\frac{w}{\sqrt{u^2+w^2}} & 0 & \frac{-u}{\sqrt{u^2+w^2}}\end{bmatrix}\begin{bmatrix}p\\q\\r\end{bmatrix}\end{aligned} \tag{E.8}$$

with A_x, A_y and A_z the specific forces.

It should be noted that the wind velocity is assumed to be either constant or with slow changing rate.

F

Adaptive Incremental Backstepping Control Design using Immersion and Invariance

F.1. Introduction

Using immersion and invariance (I&I) technique in adaptive backstepping control law can simplify the tuning process in determining the controller parameters. Similar to *tuning function* based adaptive backstepping approaches, an I&I based adaptive backstepping approach falls into the type of integrated adaptive control methods, however, the controller-related parameters of the I&I based controller is made separated from model-related parameters by using immersion and invariance. In this chapter, a nonlinear incremental controller is designed for an F-16 aircraft with the control effectiveness matrix estimated using an I&I estimator.

F.2. F-16 Motion Equations and Preliminaries on Incremental Backstepping Control

F.2.1. Motion Equations and Aerodynamic Model of F-16

For completeness purpose, the motion equations of an F-16 aircraft are given in this section. The same motion equations as those in [163] are used for the work presented in this chapter, and the reader is referred to [163] if more details are needed.

Let the controlled state vector be $\mathbf{x}_1 = \left[V_\mathbf{T}, \alpha, \beta\right]^\top$ and the second controlled state vector $\mathbf{x}_2 = \left[p_S, q_S, r_S\right]^\top$ with the subscript 'S' denoting the stability axes, then the outer loop subsystem can be represented as:

$$\dot{\mathbf{x}}_1 = f_1 + F_1 + B_1\mathbf{x}_2 + B_{1_T}\left[T\ 0\ 0\right]^\top \tag{F.1}$$

with

$$f_1 = \begin{bmatrix} g_1 \\ -p_S\tan\beta + \frac{1}{mV_\mathbf{T}\cos\beta}\left(-T\sin\alpha + mg_3\right) \\ \frac{1}{mV_\mathbf{T}}\left(-T\cos\alpha\sin\beta + mg_2\right) \end{bmatrix}$$

$$F_1 = \frac{1}{m}\begin{bmatrix} -\cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\frac{\sin\alpha}{V_\mathbf{T}\cos\beta} & 0 & \frac{\cos\alpha}{V_\mathbf{T}\cos\beta} \\ -\frac{\cos\alpha\sin\beta}{V_\mathbf{T}} & \frac{\cos\beta}{V_\mathbf{T}} & -\frac{\sin\alpha\sin\beta}{V_\mathbf{T}} \end{bmatrix}\begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix},\quad B_{1_\mathbf{T}} = \begin{bmatrix} \frac{\cos\alpha\cos\beta}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

For the inner loop, the dynamics of the subsystem are:

$$\dot{\mathbf{x}}_2 = f_2 + F_2 + B_2\left[\delta_a\ \delta_e\ \delta_r\right]^\top \tag{F.2}$$

with

$$f_2 = -\mathbf{T}_{B\to S}\mathbf{J}^{-1}\left(\omega_B \times \mathbf{J}\omega_B\right)$$

$$F_2 = \mathbf{T}_{B\to S}\mathbf{J}^{-1}\begin{bmatrix} L_0 \\ M_0 \\ N_0 \end{bmatrix},\quad B_2 = \mathbf{T}_{B\to S}\mathbf{J}^{-1}\begin{bmatrix} L_{\delta_a} & L_{\delta_e} & L_{\delta_r} \\ 0 & M_{\delta_e} & 0 \\ N_{\delta_a} & N_{\delta_e} & N_{\delta_r} \end{bmatrix} \tag{F.3}$$

where $\mathbf{T}_{B\to S}$ is the transformation matrix from body to stability axes.

F.2.2. COMPUTE THE INVERSION OF THE MOMENT OF INERTIA MATRIX

In the inversion matrix of the moment of inertia matrix J, the moment and product moment of inertia terms, i.e. i_1-i_9, are defined as follows:

$$i_1 = \frac{\left(J_{yy} - J_{zz}\right)J_{zz} - J_{xz}^2}{\tau} \qquad i_4 = \frac{J_{xz}}{\tau} \qquad i_7 = \frac{1}{J_{yy}} \tag{F.4a}$$

$$i_2 = \frac{\left(J_{xx} - J_{yy} + J_{zz}\right)J_{xz}}{\tau} \qquad i_5 = \frac{J_{zz} - J_{xx}}{J_{yy}} \qquad i_8 = \frac{\left(J_{xx} - J_{yy}\right)J_{xx} + J_{xz}^2}{\tau} \tag{F.4b}$$

$$i_3 = \frac{J_{zz}}{\tau} \qquad i_6 = \frac{J_{xz}}{J_{yy}} \qquad i_9 = \frac{J_{xx}}{\tau} \tag{F.4c}$$

where

$$\tau = J_{xx}J_{zz} - J_{xz}^2 \tag{F.5}$$

F.2.3. BASIC FORMULATION OF THE REGULAR INCREMENTAL BACKSTEPPING

Consider the second and final subsystem:

$$\dot{\mathbf{x}}_2 = \mathbf{f}(\mathbf{x}_1, \mathbf{x}_2) + G(\mathbf{x}_1, \mathbf{x}_2)\,\mathbf{u} \tag{F.6}$$

Its first order approximation is:

$$\dot{\mathbf{x}}_2 \cong \mathbf{f}(\mathbf{x}_0) + G(\mathbf{x}_0)\,\mathbf{u}_0 + \frac{\partial}{\partial x}\left[\mathbf{f}(\mathbf{x}) + G(\mathbf{x})\,\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{x}-\mathbf{x}_0) + \frac{\partial}{\partial \mathbf{u}}\left[G(\mathbf{x})\,\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} (\mathbf{u}-\mathbf{u}_0) \tag{F.7}$$

By definition, the current state rate $\dot{\mathbf{x}}_{2,0}$ satisfies:

$$\dot{\mathbf{x}}_{2,0} = \mathbf{f}(\mathbf{x}_0) + G(\mathbf{x}_0)\,\mathbf{u}_0 \tag{F.8}$$

Applying the standard linear definitions:

$$A_0 = \frac{\partial}{\partial x}\left[\mathbf{f}(\mathbf{x}) + G(\mathbf{x})\,\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} \tag{F.9}$$

$$B_0 = \frac{\partial}{\partial \mathbf{u}}\left[G(\mathbf{x})\,\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{u}=\mathbf{u}_0}} \tag{F.10}$$

Then, the following formulation is derived:

$$\dot{\mathbf{x}}_2 = \dot{\mathbf{x}}_{2,0} + A_0(\mathbf{x}-\mathbf{x}_0) + B_0\triangle\mathbf{u} \tag{F.11}$$

F.3. PRELIMINARIES ON IMMERSION AND INVARIANCE BASED ESTIMATOR

According to [8, 136], the invariant manifold is defined by:

$$\sigma = \frac{\hat{\theta}_i + \beta_i(x_i, \hat{x}) - \theta_i}{r_i} \tag{F.12}$$

where r_i are the scalar dynamic scaling parameters, and θ_i are unknown parameters.

In other words, we are using $\hat{\theta}_i + \beta_i(x_i, \hat{x})$ to approximate θ_i. The computational complexity lies in three aspects. Firstly, we need to integrate the regressor vector φ in each dimension with regard to x_i:

$$\beta_i(x_i, \hat{x}) = \gamma_i \int_0^{x_i} \varphi_i\left(\hat{x}_1, ..., \hat{x}_{i-1}, \chi, \hat{x}_{i-1}, ..., \hat{x}_n\right) d\chi \tag{F.13}$$

In other words, partial integration in multi-dimension is needed.

$$\dot{\hat{\theta}}_i = -\frac{\partial \beta_i}{\partial x_i}\left(x_{i+1} + \varphi_i(\mathbf{x})^\top\left(\hat{\theta}_i + \beta_i(x_i, \bar{x})\right)\right) - \sum_{j=1}^{n}\frac{\partial \beta_i}{\partial \hat{x}_j}\dot{\hat{x}}_j \tag{F.14}$$

Secondly, it can be seen from Eq. F.14 that the partial derivative of β_i with respect to $\hat{x}_j$ is also needed. Thirdly, the partial derivative of $\varphi_i(\mathbf{x})$ according to $\hat{x}_j$ is also required by the control law. Considering the computational complexity from all these three aspects,

it will be very complicated to design an I&I based adaptive controller if a complex linear-in-the-parameter modeling approach such as splines is chosen to construct the regressor vector φ_i.

F.4. EQUATIONS OF MOTION AND SIMPLIFIED RATE CONTROL ALLOCATION

F.4.1. INCREMENTAL EXPRESSIONS OF THE MOTION EQUATIONS

Let $\omega = [p,\ q,\ r]^\top$ being the body angular rates, then the incremental form of the body angular rate equations are as follows (the state changes are assumed to be minor during one sampling time step):

$$\dot{\omega} = \dot{\omega}_0 + \mathbf{B}_{2,\mathbf{I\&I}} \cdot \triangle\delta \tag{F.15}$$

where

$$\mathbf{B}_{2,in} = \begin{bmatrix} i_3 L_{\delta_a} + i_4 N_{\delta_a} & 0 & i_3 L_{\delta_r} + i_4 N_{\delta_r} \\ 0 & i_7 M_{\delta_e} & 0 \\ i_4 L_{\delta_a} + i_9 N_{\delta_a} & 0 & i_4 L_{\delta_r} + i_9 N_{\delta_r} \end{bmatrix} \tag{F.16}$$

with $\dot{\omega}_0$ the current measurements of the angular accelerations, $\delta = [\delta_a,\ \delta_e,\ \delta_r]^\top$, and $\triangle$ represents the incremental values over one sampling period. i_3, i_4, i_7 and i_9 are the elements of J^{-1} (see Eq. F.3), which are defined in Appendix-B. Note that $\mathbf{B}_{2,in}$ is the control effectiveness matrix defined in the body-fixed axes, which makes it slightly different from B_2 matrix shown in Eq. F.3.

In order to simplify the model estimation, the $B_{2,in}$ matrix is further simplified by eliminating the non-dominant coefficients (e.g.$i_4 N_{\delta_a}$):

$$\mathbf{B}_{2,\mathbf{I\&I}} = \begin{bmatrix} i_3 L_{\delta_a} & 0 & i_3 L_{\delta_r} \\ 0 & i_7 M_{\delta_e} & 0 \\ i_9 N_{\delta_a} & 0 & i_9 N_{\delta_r} \end{bmatrix} \tag{F.17}$$

This system shown by Eq. F.15 can be rewrien into a standard form to facilitate the design procedure of the Immersion and Invariance (I&I) estimator:

$$\dot{p} = f_4 + \varphi_4^\top \theta_4 \tag{F.18a}$$
$$\dot{q} = f_5 + \varphi_5^\top \theta_5 \tag{F.18b}$$
$$\dot{r} = f_6 + \varphi_6^\top \theta_6 \tag{F.18c}$$

with $f_4 = \dot{p}_{b,0}$, $f_5 = \dot{q}_{b,0}$ and $f_6 = \dot{r}_{b,0}$ are the measurements of the angular accelerations. If we suppose that each element of $\mathbf{B}_{2,\mathbf{I\&I}}$ is a constant (i.e. the function of the incremental inputs is a linear function), then:

$$\varphi_4 = [i_3 \triangle\delta_a,\ i_3 \triangle\delta_r]^\top \tag{F.19a}$$
$$\varphi_5 = i_7 \triangle\delta_e \tag{F.19b}$$
$$\varphi_6 = [i_9 \triangle\delta_a,\ i_9 \triangle\delta_r]^\top \tag{F.19c}$$

and

$$\theta_4 = \left[L_{\delta_a},\ L_{\delta_r}\right]^\top \tag{F.20a}$$
$$\theta_5 = M_{\delta_e} \tag{F.20b}$$
$$\theta_6 = \left[N_{\delta_a},\ N_{\delta_r}\right]^\top \tag{F.20c}$$

F.5. IMMERSION AND INVARIANCE ESTIMATE OF THE CONTROL EFFECTIVENESS MATRIX

F.5.1. DESIGN OF THE IMMERSION AND INVARIANCE ESTIMATOR

L_{δ_a} from Eq. F.17 can be expressed as follows:

$$L_{\delta_a} = \bar{q}Sb \cdot C_{L_{\delta_a}} \tag{F.21}$$

with S the wing area, $\bar{q}$ the dynamic pressure ratio and b the wing span. Therefore, Eq. F.19 can be changed into:

$$\varphi_4 = \left[i_3\bar{q}Sb\cdot\Delta\delta_a,\ i_3\bar{q}Sb\cdot\Delta\delta_r\right]^\top \tag{F.22a}$$
$$\varphi_5 = i_7\bar{q}S\bar{c}\cdot\Delta\delta_e \tag{F.22b}$$
$$\varphi_6 = \left[i_9\bar{q}Sb\cdot\Delta\delta_a,\ i_9\bar{q}Sb\cdot\Delta\delta_r\right]^\top \tag{F.22c}$$

Correspondingly, the unknown parameters shown in Eq. F.20 turn into:

$$\theta_4 = \left[C_{L_{\delta_a}},\ C_{L_{\delta_r}}\right]^\top \tag{F.23a}$$
$$\theta_5 = C_{M_{\delta_e}} \tag{F.23b}$$
$$\theta_6 = \left[C_{N_{\delta_a}},\ C_{N_{\delta_r}}\right]^\top \tag{F.23c}$$

According to the I&I estimation theory, the estimator takes the following form:

$$\beta_4 = \left[\beta_{C_{L_{\delta_a}}},\ \beta_{C_{L_{\delta_r}}}\right]^\top = \left[\gamma_4 i_3\bar{q}Sb\Delta\delta_a p,\ \gamma_4 i_3\bar{q}Sb\Delta\delta_r p\right]^\top \tag{F.24}$$

In further:

$$C_{L_{\delta_a}} = \hat{C}_{L_{\delta_a}} + \beta_{C_{L_{\delta_a}}} \tag{F.25}$$

Using Eq. F.24, the update law for $\beta_{C_{L_{\delta_a}}}$ takes the following form:

$$\dot{\hat{C}}_{L_{\delta_a}} = -\gamma_4 i_3\bar{q}Sb\cdot\Delta\delta_a\cdot\dot{p} - \gamma_4 i_3\bar{q}Sb\cdot p\cdot\Delta\dot{\delta}_a \tag{F.26}$$

The update laws for other 4 control derivatives,i.e. $\dot{\hat{C}}_{L_{\delta_r}}$,$\dot{\hat{C}}_{M_{\delta_e}}$,$\dot{\hat{C}}_{N_{\delta_a}}$ and $\dot{\hat{C}}_{N_{\delta_r}}$, can be derived in the same way.

$$\dot{\hat{C}}_{L_{\delta_r}} = -\gamma_4 i_3\bar{q}Sb\cdot\Delta\delta_r\cdot\dot{p} - \gamma_4 i_3\bar{q}Sb\cdot p\cdot\Delta\dot{\delta}_r \tag{F.27}$$

F

$$\dot{\hat{C}}_{M_{\delta_e}} = -\gamma_5 i_7 \bar{q} S \bar{c} \cdot \Delta\delta_e \cdot \dot{q} - \gamma_5 i_7 \bar{q} S \bar{c} \cdot q \cdot \Delta\dot{\delta}_e \tag{F.28}$$

$$\dot{\hat{C}}_{N_{\delta_a}} = -\gamma_6 i_9 \bar{q} S b \cdot \Delta\delta_a \cdot \dot{r} - \gamma_6 i_9 \bar{q} S b \cdot r \cdot \Delta\dot{\delta}_a \tag{F.29}$$

$$\dot{\hat{C}}_{N_{\delta_a}} = -\gamma_6 i_9 \bar{q} S b \cdot \Delta\delta_r \cdot \dot{r} - \gamma_6 i_9 \bar{q} S b \cdot r \cdot \Delta\dot{\delta}_r \tag{F.30}$$

As can be seen from Eqs.F.26-F.30, time derivatives of control surface deflections and time derivatives of body angular rates are needed. Thereafter, using Eq. F.25 and then Eq. F.17, the control effectiveness matrix can be calculated.

F.5.2. GENERALIZED EXPRESSION OF φ AND β IN CASE OF USING COMPLEX REGRESSORS

In this section, a generalized expression of the terms related to the regressor vector φ when designing an I&I Estimator is given in case that model identification algorithm with a more complex model structure is used to construct the regressor vector φ. If we assume the elements in the control effectiveness matrix is time varying, and is related to some states, then Eq. F.22 can be modified as follows:

$$\varphi_4 = i_3 \bar{q} S b \cdot \left[\cdot \Delta\delta_a \cdot \mathscr{S}_1(\beta,\, p),\ \Delta\delta_r \cdot \mathscr{S}_2(\beta,\, p)\right]^\top \tag{F.31a}$$

$$\varphi_5 = i_7 \bar{q} S \bar{c} \cdot \Delta\delta_e \cdot \mathscr{S}_3(\alpha,\, q) \tag{F.31b}$$

$$\varphi_6 = i_9 \bar{q} S b \cdot \left[\cdot \Delta\delta_a \cdot \mathscr{S}_4(\beta,\, p),\ \Delta\delta_r \cdot \mathscr{S}_5(\beta,\, p)\right]^\top \tag{F.31c}$$

Integrating Eq.19a, Eq. F.24 becomes:

$$\beta_4 = \left[\beta_{C_{L_{\delta_a}}},\, \beta_{C_{L_{\delta_r}}}\right]^\top = \gamma_4 i_3 \bar{q} S b \cdot \left[\Delta\delta_a \cdot \mathscr{S}_1(\beta,\, p)\big|_{\text{int}}^{p},\ \Delta\delta_r \cdot \mathscr{S}_2(\beta,\, p)\big|_{\text{int}}^{p}\right]^\top \tag{F.32a}$$

$$\beta_5 = \beta_{C_{M_{\delta_a}}} = \gamma_5 i_7 \bar{q} S \bar{c} \cdot \Delta\delta_a \cdot \mathscr{S}_3(\alpha,\, q)\big|_{\text{int}}^{q} \tag{F.32b}$$

$$\beta_6 = \left[\beta_{C_{N_{\delta_a}}},\, \beta_{C_{N_{\delta_r}}}\right]^\top = \gamma_6 i_9 \bar{q} S b \cdot \left[\Delta\delta_a \cdot \mathscr{S}_4(\beta,\, p)\big|_{\text{int}}^{p},\ \Delta\delta_r \cdot \mathscr{S}_5(\beta,\, p)\big|_{\text{int}}^{p}\right]^\top \tag{F.32c}$$

F.6. RESULTS AND ANALYSIS

F.6.1. TUNING GAINS

Table F.1: The control gains of the I&I based IBKS controller, $k_{p_{V_{\text{TAS}}}} = 2.5$

$\gamma_4 = 0.006\ rad/s$	$k_{p_\phi} = 5$	$k_{i_\phi} = 0.5$	$k_{p_p} = 5$
$\gamma_5 = 0.05\ rad/s$	$k_{p_\alpha} = 3.5$	$k_{i_\alpha} = 0$	$k_{p_q} = 1.5$
$\gamma_6 = 0.008\ rad/s$	$k_{p_\beta} = 2$	$k_{i_\beta} = 0$	$k_{p_r} = 5$

Controller gains are given in Table F.1.

F.6.2. ANGULAR TRACKING PERFORMANCE

In this section, the regular incremental backstepping (reg-IBKS) control approach is compared with the singular perturbation based IBKS (spt-IBKS). A tracking controller for the true airspeed, angle of attack, sideslip angle, and roll rate in the stability axis is designed for an F-16 aircraft and validated. In the regular IBKS controller, the control effectiveness matrix is updated in real-time using the I&I estimator designed in section F.5.2.The simulation results showing the reference command tracking performance of the controller are plotted in Figs.F.1-fig:ii06. The commands for the roll angle and angle of attack are given sequentially. As shown in these 6 figures, spt-IBKS nearly achieves exactly the same command tracking performance as that achieved by a regular IBKS controller.

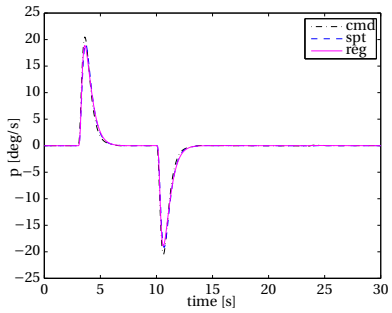


Figure F.1: Roll rate, IBKS.

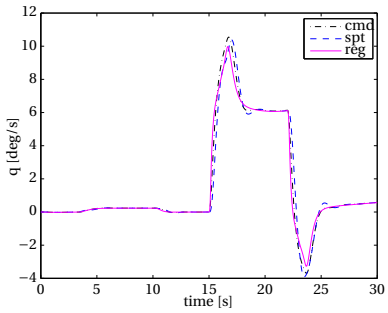


Figure F.2: Pitch rate, IBKS.

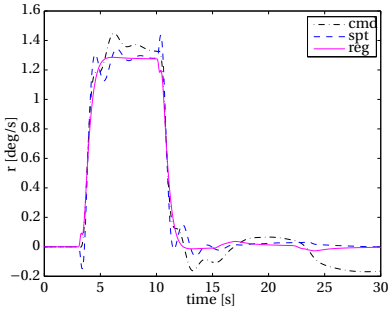


Figure F.3: Yaw rate, IBKS.

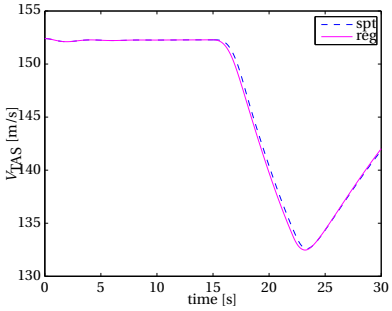


Figure F.4: True airspeed, IBKS.

F.6.3. IMMERSION AND INVARIANCE BASED ESTIMATION PERFORMANCE

In Section F.6.2, only the tracking performances of the controlled variables are given, the time-varying histories of the control derivatives, i.e. elements in the control effectiveness matrix, are given and analyzed in this section.

The estimation results of the control derivatives when using an I&I estimator are plotted in Figs.F.8-F.12. According to these 5 figures, the changing trends of the control

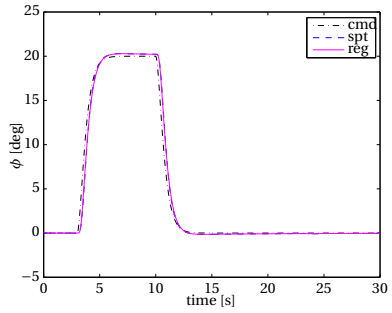


Figure F.5: Roll angle, IBKS.

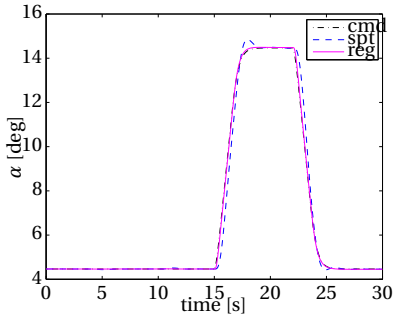


Figure F.6: Angle of attack, IBKS.

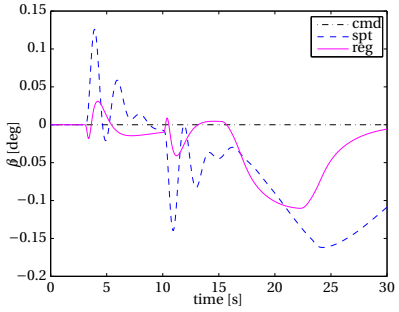


Figure F.7: Sideslip angle, IBKS.

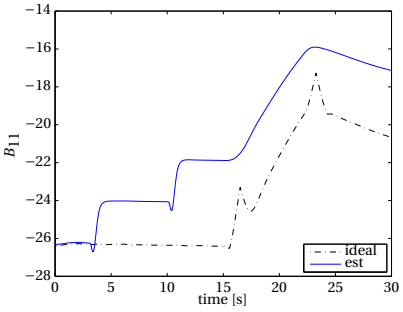


Figure F.8: B_{11}, IBKS.

derivatives related to ailerons and rudders are well reflected, but the estimation of the elevator related control derivatives are not satisfied.

F.7. CONCLUSIONS

This chapter investigated how the immersion and invariance I&I technique can be applied to provide an estimation of the control effectiveness matrix when designing an incremental backstepping (IBKS) controller for an F-16 aircraft. The control performance of a regular IBKS is compared with that of a singular perturbation based incremental backstepping control law namely sensor based backstepping with a focus on investigating how the accuracy of the control effectiveness matrix affect the control performance of an incremental backstepping controller. Closed-loop simulation results show that the IBKS is not sensitive to the parameter variations in the control effectiveness matrix since the control performances of an IBKS controller do not have apparent difference before and after including parameter adaption of the control derivatives. This should be due to the fact that the chosen flight scenarios of an F-16 aircraft are located in a moderate range of its flight envelopes. Further applications on re-entry vehicles which has a larger parameter variations than aerial vehicles should be investigated.

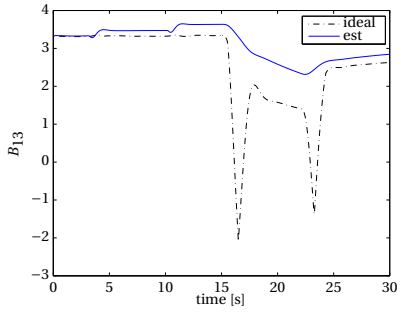


Figure F.9: B_{13}, IBKS.

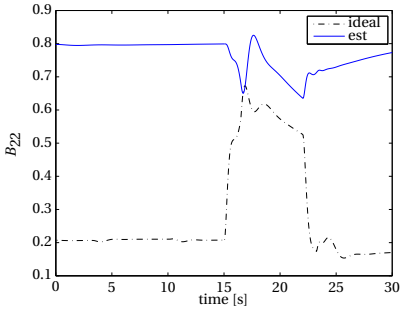


Figure F.10: B_{22}, IBKS.

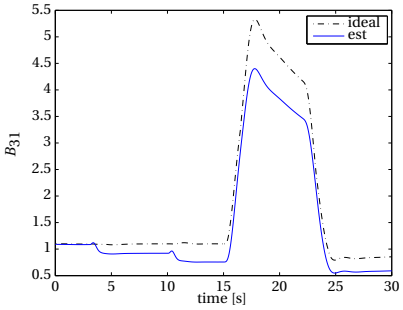


Figure F.11: B_{31}, IBKS.

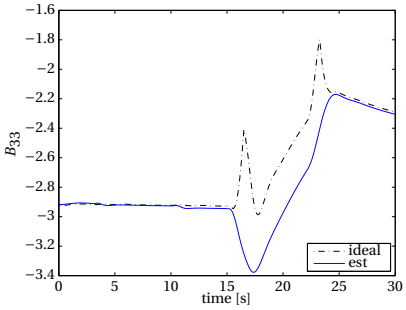


Figure F.12: B_{33}, IBKS.

G

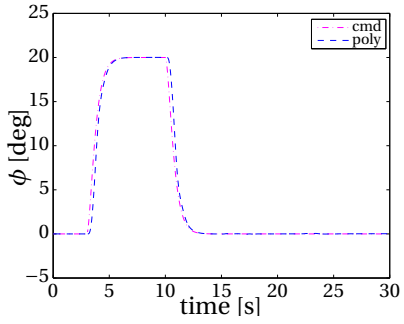
Parameters of an NDI Controller for Closed-Loop Data Collection in Validating TPS B-splines

A regular polynomial basis based adaptive nonlinear dynamic inversion (NDI) controller has been designed for the F-16 fighter aircraft. The natural frequency of the second order command filters, see.[47]), and the controller gains are given in Table G.1. In Table G.1, the subscript 'I' denotes integrator related terms, i.e. integration gains, ω represents the natural frequency of the command filters. For more details of the adaptive NDI control approach, the reader is referred to Lombaerts et al. [86][p.222-232].

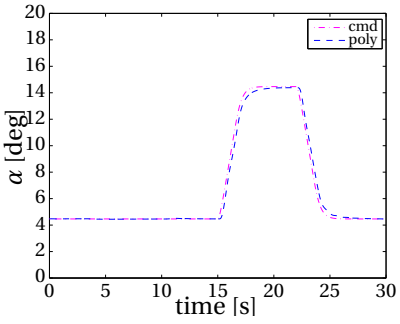
Table G.1: The natural frequency of the command filters and control gains of the adaptive NDI controller

$\omega_\phi = 2.5\ rad/s$	$\omega_p = 30\ rad/s$	$k_\phi = 5$	$k_{\phi_I} = 0$	$k_p = 10$
$\omega_\alpha = 3\ rad/s$	$\omega_q = 30\ rad/s$	$k_\alpha = 8$	$k_{\alpha_I} = 5$	$k_q = 5$
$\omega_\beta = 6\ rad/s$	$\omega_r = 30\ rad/s$	$k_\beta = 1$	$k_{\beta_I} = 0.05$	$k_r = 1.6$

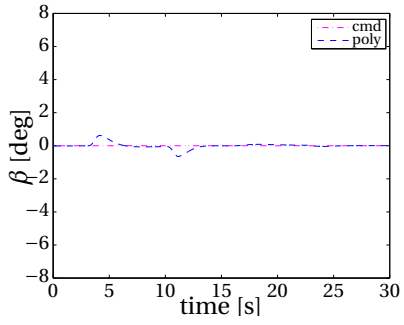
During the data collection of the second validation data set, i.e. the closed-loop data set shown in Section 8.5.3, the step-type angular reference commands are given to the closed-loop F-16 aircraft. The responses and changing history of the attitude angles and the body angular rates are shown in Figure G.1. In this figure, 'cmd' denotes the given command, and 'poly' represents the responses of the controlled system with the controller designed using the polynomial basis based adaptive NDI controller.



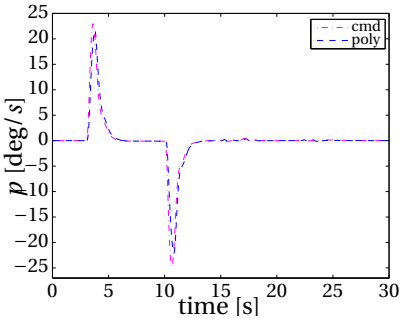
(a) Roll angle.



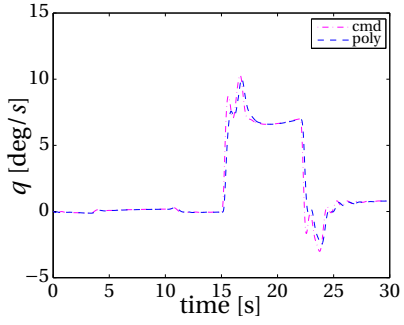
(b) Angle of attack.



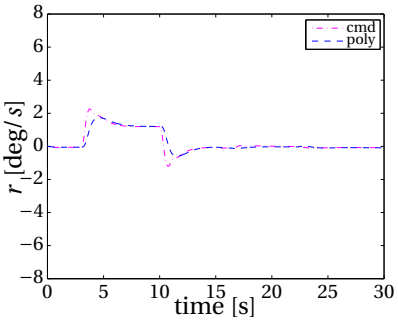
(c) Sideslip control.



(d) Roll rate.



(e) Pitch rate.



(f) Yaw rate.

Figure G.1: Angular and angular rate changes of the F-16 aircraft armed with a polynomial based adaptive NDI controller.

H

PRELIMINARIES ON SINGULAR PERTURBATION THEORY AND TIKHONOV'S THEOREM

H.1. SINGULAR PERTURBATION THEORY, SPT

Singular Perturbation Theory [59, 67]:
A singularly perturbed, nonlinear system with the following system description is considered:

$$\dot{x} = f(t,x,u,\epsilon),\ x(0) = \xi(\epsilon) \tag{H.1a}$$

$$\epsilon\dot{u} = g(t,x,u,\epsilon),\ u(0) = \eta(\epsilon) \tag{H.1b}$$

where ξ and η depend smoothly on the small positive parameter ϵ Assume that f and g are continuously differentiable in their arguments for $(t,x,u,\epsilon) \in \mathbb{R}^+ \times D_x \times D_u \times [0,\epsilon_0]$, where $D_x \in \mathbb{R}^n$, $D_u \in \mathbb{R}^m$ are domains and $\epsilon_0 > 0$. In addition, it is assumed that $g(t,x,u,0) = 0$ has $k \geq 1$ isolated real roots $u = h_i(t,x)$, $i \in \{1,...,k\}$ for each $(t,x) \in \mathbb{R}^+ \times D_x$. The i-th model for a reduced (slow) system can be obtained by substituting the roots into Eq. H.1 at $\epsilon = 0$:

$$\dot{x} = f(t,x,h(t,x),0),\ x(0) = \xi(0) \tag{H.2}$$

In singular perturbation theory, the boundary layer (fast) system is given by:

$$\frac{dv}{d\tau} = g(t,x,v+h(t,x),0),\ v(0) = \eta(0) - h(0,\xi_0) \tag{H.3}$$

where $v(t,x) = u - h(t,x)$, the normal time-scale t, in Eq. H.1, is replaced by the new time-scale $\tau = \frac{t}{\epsilon}$, $\eta_0 = \eta(0)$ and $\xi_0 = \xi(0)$, $(t,x) \in \mathbb{R}^+ \times D_x$ are treated as fixed parameters.

H.2. TIKHONOV'S THEOREM

Tikhonov's Theorem [59, 67, 153]:
Consider the singularly perturbed system given in Eq. H.1, and let $u = h(t,x)$ be an isolated root of $g(t,x,u,0)$. Assume that the following conditions hold for all $(t,x,u,\epsilon) \in \mathbb{R}^+ \times D_x \times D_u \times [0,\epsilon_0]$ for the domains $D_x \in \mathbb{R}^n$, $D_u \in \mathbb{R}^m$, which contain the corresponding origins.

(a) On any compact subset of $D_x \times D_v$ the function f, g, their first partial derivatives with respect to (x,u,ϵ), and the first partial derivative of g with respect to t are continuous and bounded, $h(t,x)$ and $\frac{\partial g}{\partial u}(t,x,u,0)$ have bounded first derivatives with respect to their arguments, $\frac{\partial f}{\partial x}(t,x,h(t,x),0)$ is Lipschitz in x uniformly in t, and the initial conditions for ξ and η are smooth functions of ϵ.

(b) The origin is an exponentially stable equilibrium point of the reduced system given in Eq. H.2. There exists a Lyapunov function V: $\mathbb{R}^+ \times D_x \mapsto \mathbb{R}^+$ that satisfies:

$$W_1(x) \leq V(t,x) \leq W_2(x) \tag{H.4a}$$

$$\frac{\partial V(t,x)}{\partial t} + \frac{\partial V(t,x)}{\partial x} f(t,x,h(t,x),0) \leq -W_3(x) \tag{H.4b}$$

for all $(t,x) \in \mathbb{R}^+ \times D_x$, where W_1, W_2, and W_3 are continuous positive definite functions on D_x. Let c be a nonnegative number such that $\{x \in D_x | W_1(x) \leq c\}$ is a compact subset of D_x.

(c) The origin is an equilibrium point of the boundary layer system given in Eq. H.3, which is exponentially stable uniformly in (t,x).

Let $R_v \in D_v$ denote the region of attraction of the autonomous system:

$$\frac{dv}{d\tau} = g(0,\xi_0, v + h(0,\xi_0),0) \tag{H.5}$$

and let W_v be a compact subset of R_v. Then for each compact set $W_x \in \{x \in D_x | W_2(x) \leq \rho c, \ 0 < \rho < 1\}$, there exists a positive constant ϵ_* such that for all $t \geq 0$, $\xi_0 \in W_x$, $\big(\eta_0 - h(0,\eta_0)\big) \in W_v$ and $0 < \epsilon < \epsilon_*$ such that the system given by Eq. H.1 has a unique solution x_ϵ on $\mathbb{R}^+$ and

$$x_\epsilon(t) - x_{00}(t) = O(\epsilon) \tag{H.6}$$

holds uniformly for $t \in \mathbb{R}^+$, where $x_{00}(t)$ denotes the solution of the reduced system given in Eq. H.2.

H.3. APPROXIMATE DYNAMIC INVERSION, ADI

A controller called approximate dynamic inversion was developed by Hovakimyan et al.[59] for a nonlinear non-affine in control system based on singular perturbation theory and Tikhonov's theorem. The system considered here is:

$$\dot{x} = f(t,x,u) \tag{H.7}$$

where $x(0) = x_0$ for (x,u) and where $D_x \subset \mathbb{R}^n$ and $D_u \subset \mathbb{R}^m$ are domains that contain the corresponding origins. Here x denotes the state vector and u the input vector. The function f is continuously differentiable in its arguments. Furthermore assume $\partial f/\partial u$ is bounded away from zero for $(x,u) \in \Omega_{x,u} \subset D_x \times D_u$, where $\Omega_{x,u}$ is a compact set of possible initial conditions, i.e., there exists a $b_0 > 0$ such that $|\partial f/\partial u| > b_0$

A reference signal y_r is introduced for the state x, and the track error is defined as $e = x - y_r$. Then the error dynamics of the system can be written as [44]:

$$\dot{e} = f\left(t, e + y_r, u\right) - \dot{y}_r \tag{H.8}$$

An approximate dynamic inversion controller for Eq. H.8 can be designed using the following fast dynamic approximation:

$$\epsilon \dot{u} = -sign\left(\frac{\partial f}{\partial u}\right)\mathbf{f}(t,e,u) \tag{H.9}$$

Remarks: If we use backstepping to design a controller for Eq. H.7, Eq. H.8 can have the following formulation:

$$\dot{e} = f\left(t, e + y_r, u\right) - \dot{y}_r = -a \cdot e \tag{H.10}$$

where a is a controller gain, and is a positive constant. Then the derived algebraic equation related to $f\left(t, e + y_r, u\right)$ can be solved further using a fast-approaching function given by Eq. H.9.

Theorem of Approximate Dynamic Inversion [59]:
Assume that the following conditions hold for all $(t, e, u - h(t,e), \epsilon) \in \mathbb{R}^+ \times D_e \times D_v \times [0, \epsilon_0]$ for some domains $D_e \in \mathbb{R}^n$ and $D_v \in \mathbb{R}^m$, which contain the corresponding origins.

(a) On any compact subset of $D_e \times D_v$ the function f and the first partial derivatives with respect to (t,e,u) are continuous and bounded, $h(t,e)$ and $\frac{\partial f}{\partial u}(t,e,u)$ have bounded first derivatives with respect to their arguments, $\frac{\partial f}{\partial e}(t,e,h(t,e))$ is Lipschitz in e, uniformly in t.

(b) The origin is an exponentially stable equilibrium point of the reduced system

$$\dot{e} = f\left(t, y_r, h(t,e)\right) \tag{H.11}$$

The mapping $e \mapsto f\left(e + y_r, h(t,e)\right)$ is continuously differentiable and Lipschitz in e, uniformly in t.

(c) The mapping $(t,e,v) \mapsto \frac{\partial f}{\partial u}(t,e,v + h(t,e))$ is bounded from below by some positive number for all $(t,e) \in \mathbb{R}^+ \times D_e$.

Then the origin of the boundary layer system

$$\frac{dv}{d\tau} = -sign\left(\frac{\partial f}{\partial u}\right)\mathbf{f}(t,e,v + h(t,e)) \tag{H.12}$$

is exponentially stable. Moreover, let Ω_v be a compact subset of R_v, where $R_v \subset D_v$ denotes the region of attraction of the following autonomous system

$$\frac{dv}{d\tau} = -sign\left(\frac{\partial f}{\partial u}\right)\mathbf{f}(0, e_0, v + h(0, e_0)) \tag{H.13}$$

Then for each compact subset $\Omega_e \subset D_e$, there exists a positive constant ϵ_* such that for all $t \geq 0$, $e_0 \in \Omega_e$, $u_0 - h(0, e_0) \in \Omega_v$, and $0 < \epsilon < \epsilon_*$. The nonlinear system, Eq. H.7, has a unique solution x_ϵ on $\mathbb{R}^+$ and $x_\epsilon(t) = y_r(t) + O(\epsilon)$ holds uniformly for $t \in [t, +\infty)$.

H.4. RELATIONS BETWEEN ADI, SPT AND TIKHONOV'S THEOREM

Firstly, the ADI controller developed by Hovakimyan et al. [59], see Eqs. H.8-H.9, introduced new system dynamics due to the dynamic approximation control algorithm given by Eq. H.9. The dynamics introduced by the controller approximation process are viewed as fast dynamics if the controlled plant is viewed as a singularly perturbed system.

Secondly, Hovakimyan et al. [59] chose a fast-approaching function Eq. H.9, which takes the form of Eq. H.5, such that Tikhonov's theorem given in Section H.2 can be applied to derive a controller to stabilize the overall controlled system.

Thirdly, to use Tikhonov's theorem to design a stable controller for a non-affine in control nonlinear system, the fast approximation function given by Eq. H.9 is not the only option for Eq. H.5. Furthermore, the $sign(\cdot)$ function in Eq. H.9 may sacrifice the optimality of the controller. It is more likely that the final controller could be made more optimal without sacrificing the stability property if the function $sign\left(\frac{\partial f}{\partial u}\right)$ in Eq. H.9 is replaced by the Jacobian matrix $\frac{\partial f}{\partial u}$ or the entire fast-approaching function given by Eq. H.9 is replaced by the other proper functions.

REFERENCES

[1] Acquatella, P., Kampen, C. C. van, and Chu, Q. Incremental backstepping for robust nonlinear flight control. In *EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation and Control*, Delft, The Netherlands, 2013. CEAS.

[2] Ait Haddou Ali, A., Chu, Q., Kampen, E. van, and Visser, C. C. de. Exploring adaptive incremental backstepping using immersion and invariance for an f-16 aircraft. In *AIAA Guidance, Navigation and Control Conference*, Maryland, 2014. AIAA Paper:2014-0084.

[3] Allen, R. C., Kwatny, H. G., and Bajpai, G. Safe set protection and restoration for unimpaired and impaired aircraft. In *AIAA Guidance, Navigation and Control Conference*, Minneaplis, Minnesota, 2012. AIAA Paper: 2012-4822.

[4] Alwardi, H., Wang, S., Jennings, L. S., and Richardson, S. An adaptive least-squares collocation radial basis function method for the hjb equation. *Journal of Global Optimization*, 52(2):305–322, 2012.

[5] Alwi, H. *Fault Tolerant Sliding mode Control Schemes with Aerospace Applications.* PhD thesis, University of Leicester, England, Feb. 2008.

[6] Alwi, H., Edwards, C., Stroosma, O., and Mulder, J. A. Evaluation of a sliding mode fault-tolerant flight controller for the el al incident. *Journal of Guidance, Control and Dynamics*, 33(3):677–694, 2010. doi:10.2514/1.47442.

[7] An, S., Liu, W., and Venkatesh, S. Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression. *Pattern Recognition*, 40(8): 2154–2162, 2007. doi:10.1016/j.patcog.2006.12.015.

[8] Astolfi, A., Karagiannis, D., and Ortega, R. *Nonlinear and Adaptive Control with Applications.* Springer, 2008.

[9] Awanou, G. and Lai, M. J. On convergence rate of the augmented lagrangian algorithm for nonsymmetric saddle point problems. *Applied Numerical Mathematics*, 54:122–134, 2005. doi:10.1016/j.apnum.2004.09.020.

[10] Awanou, G., Lai, M. J., and Wenston, P. *The Multivariate Spline Method for Scattered Data Fitting and Numerical Solutions of Partial Differencial Equations.* Wavelets and Splines, 2005. pp.24-75.

[11] Bardi, M. and Capuzzo-Dolcetta, I. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations.* Birkhäuser Mathematics, Boston, 2008.

[12] Belcastro, C. M. Loss of control prevention and recovery: Onboard guidance, control, and system technologies. In *AIAA Guidance, Navigation and Control Conference*, Minneapolis, Minnesota, 2012. AIAA Paper:2012-4762.

[13] Bernhard, S., Sung, K.-K., Charis, J. C., and Burges, F. G. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11):2758–2765, 1997.

[14] Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific optimization and computation series, Nashua, New Hampshire, 1996.

[15] Bi, J., Zhang, T., and Bennett, K. Column-generation boosting methods for mixture of kernels. In *Proceeding of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 521–526, New York, 2004. KDD.

[16] Billings, S. A. and Chen, S. The identification of linear and non-linear models of a turbocharged automotive diesel engine. *Mech. Syst. Signal Process*, 3(2):123–142, 1989.

[17] Bodson, M. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control and Dynamics*, 25(4):703–711, 2002. doi:10.2514/2.4937.

[18] Boeing. Statistical summary of commercial jet airplane accidents-worldwide operations 1959-2012. Technical report, Boeing Commercial Airplanes, United States, 2012. available at: http://www.boeing.com/news/techissues/pdf/statsum.pdf.

[19] Boskovic, J. D., Dongmo, J.-E. T., and Knoebel, N. An adaptive fault management system for resilient flight control. In *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, 2009. AIAA Paper: 2009-6263.

[20] Botella, O. and Shariff, K. B-spline methods in fluid dynamics. *International Journal of Computational Fluid Dynamics*, 17(2):133–149, 2010. doi:10.1080/1061856031000104879.

[21] Bottou, L., Chapelle, O., DeCoste, D., and Weston, J. *Large-Scale Kernel Machines*. MIT Press, 2006.

[22] Brinker, J. S. and Wise, K. A. Flight testing of reconfigurable control law on the x-36 tailless aircraft. *Journal of Guidance, Control and Dynamics*, 24(5):903–909, 2001. doi:10.2514/2.4826.

[23] Burcham, F. W., Maine, T. A., and Burken, J. J. Using engine thrust for emergency flight control:md-11 and b-747 results. Nasa/tm-1998-206552, NASA Dryden Flight Research Center, May 1998.

[24] Cawley, G. and Talbot, N. L. C. Reduce rank kernel ridge regression, letter. *Neural Process*, 16(3):293–302, 2002.

[25] Charlet, B., Levine, J., and Mario, R. On dynamic feedback linearization. *System and Control Letters*, 13:143–152, 1989.

[26] Chen, S., Cowan, C. F. N., and Grant, P. M. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on neural networks*, 2(2):302–309, 1991.

[27] Chowdhary, G. V., Mühlegg, M., and Johnson, E. N. Exponential parameter and tracking error convergence guarantees for adaptive controllers without persistency of excitation. *International Journal of Control*, 87(8):1583–1603, 2014. doi:10.1080/00207179.2014.880128.

[28] Christopher, M. B. *Pattern Recognition and Machine Learning*. Springer, 2006.

[29] Das, S., Kuttieri, R. A., Sinha, M., and Jategaonkar, R. Neural partial differential method for extracting aerodynamic derivatives from flight data. *Journal of Guidance, Control and Dynamics*, 33(2):376–384, 2010. doi:10.2514/1.46053.

[30] Boor, C. de. B-form basics. In *Geometric modeling: algorithms and new trends, edited by Farin, G.*, Philadelphia, USA, 1987. Society for Industrial and Applied Mathematics. pp.131-148.

[31] De Brabanter, K., De Brabanter, J., Suykens, J. A. K., and De Moor, B. Optimized fixed-size kernel models for large data sets. *Computa. Stat. Data Anal.*, 54(6):1484–1504, 2010.

[32] De Kruif, B. J. and De Vries, T. J. A. Pruning error minimization in least squares support vector machines. *IEEE Trans. Neural Networks*, 14(3):696–702, 2004.

[33] Visser, C. de. *Global Nonlinear Model Identification with Multivariate Splines*. PhD thesis, Delft University of Technology, The Netherlands, July 2011.

[34] Visser, C. C. de, Chu, Q. P., and Mulder, J. A. A new approach to linear regression with multivariate splines. *Automatica Journal*, 45:2903–2909, 2009. doi:10.1016/j.automatica.2009.09.017.

[35] Visser, C. C. de, Mulder, J. A., and Chu, Q. Global nonlinear aerodynamic model identification with multivariate splines. In *AIAA Atomospheric Flight Mechanics Conference*, Chicago, Illinois, 2009. AIAA.

[36] Visser, C. C. de, Mulder, J. A., and Chu, Q. A multidimensional spline based global nonlinear aerodynamic model for the cessna citation ii. In *AIAA Atomospheric Flight Mechanics Conference*, Toronto, Ontario Canada, 2010. AIAA Paper:2010-7950.

[37] Visser, C. C. de, Chu, Q. P., and Mulder, J. A. Differential constraints for bounded recursive identification with multivariate splines. *Automatica Journal*, 47:2059–2066, 2011. doi:10.1016/j.automatica.2011.06.011.

[38] Dunbar, B. Intelligent flight control system. Nasa dryden fact sheet, Dryden Flight Research Center, Edwards, California, Apr. 2010.

[39] Durham, W. C. Constrained control allocation. *Journal of Guidance, Control and Dynamics*, 16(4):717–725, 1993. doi:10.2514/3.21072.

[40] Durham, W. C. Constrained control allocation. *Journal of Guidance, Control and Dynamics*, 16(4):717–725, 1993. doi:10.2514/3.21201.

[41] Edwards, C., Lombaerts, T. J. J., and Smaili, H. *Fault Tolerant Flight Control - A Benchmark Challenge*. Springer, 2010. 1st ed. Lecture Notes in Control and Information Science.

[42] Egbert, L. and Halley, I. Stabilator reconfiguration flight testing on the f/a-18e/f. In *Proceedings of the SAE Control and Guidance Meeting*, 2001. Mar.

[43] Eslinger, R. and Chandler, P. Self-repairing flight control system program overview. In *Proceedings of IEEE 1988 National Aerospace and Electronics Conference*, pages 504–511, Dayton, Ohio, 1988. IEEE.

[44] Falkena, W. *Investigation of Practical Flight Control Systems for Small Aircraft*. PhD thesis, Delft University of Technology, The Netherlands, Dec. 2012. pp.131-158.

[45] Falkena, W., Oort, E. van, and Chu, Q. Towards certifiable advanced flight control systems, a sensor based backstepping approach. In *AIAA Guidance, Navigation and Control Conference*, Portland, Oregon, 2011. AIAA Paper 2011-6482.

[46] Falkena, W., Borst, C., Oort, E. R. van, and Chu, Q. P. Sensor-based backstepping. *Journal of Guidance, Control and Dynamics*, 36(2):606–610, 2013. doi:10.2514/1.56581.

[47] Farrell, J., Sharma, M., and Polycarpou, M. Backstepping-based flight control with adaptive function approximation. *Journal of Guidance, Control and Dynamics*, 28 (6):1089–1102, 2005. doi:10.2514/1.13030.

[48] Fuller, J. W. Integrated flight propulsion control for loss-of-control prevention. In *AIAA Guidance, Navigation and Control Conference*, Minneapolis, Minnesota, 2012. AIAA Paper 2012-4896.

[49] Gavrilets, V. Damage tolerant flight control systems for unmanned aircraft. In *8th ICAS/ATIO*, Anchorage, Alaska, 2008.

[50] Gilyard, G. B., Conley, J. L., Le, J., and Burcham, J. F. W. A simulation evaluation of a four-engine jet transport using engine thrust modulation for flightpath control. In *AIAA Guidance, Navigation and Control Conference*, Toronto, Ontario Canada, 1991. AIAA Paper 1991-2223.

[51] Girich, C. and Ravindra, J. Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter. *Journal of Aerospace Science and Technology*, 14(2):106–117, 2010. doi:10.1016/j.ast.2009.10.003.

[52] Goman, M., Khramtsovsky, A., and Kolesnikov, E. Evaluation of aircraft performance and maneuverbility by computation of attainable equilibrium sets. *Journal of Guidance, Control and Dynamics*, 31(2):329–339, 2008. doi:10.2514/1.29336.

[53] Goupil, P. Oscillatory failure case detection in the a380 electrical flight control system by analytical redundancy. *Control Engineering Practice*, 18:1110–1119, 2010. DOI:10.1016/j.conengprac.2009.04.003.

[54] Govindarajan, N., Visser, C. C. de, and Krishnakumar, K. A sparse collocation method for solving time-dependent hjb equations using multivariate b-splines. *Automatica Journal*, 50:2234–2244, 2014. doi:10.1016/j.automatica.2014.07.012.

[55] Härkegård, O. *Backstepping and Control Allocation with Applications to Flight Control*. PhD thesis, Linköping University, Sweden, Apr. 2003.

[56] Hoegaerts, L., Suykens, J. A. K., Vandewalle, J., and De Moor, B. A comparison of pruning algorithms for sparse least squares support vector machines. In *International Conference on Neural Information Processing*, pages 1247–1253, Calcutta, India, 2004. Springer.

[57] Holzapfel, F. *Nichtlineare Adaptive Regelung Eines Unbemannten Fluggerätes*. PhD thesis, Technische Universit"at München, Germany, May 2004. Lehrstuhl für Flugmechanik und Flugregelung.

[58] Holzapfel, F., Heller, M., Weingartner, M., Sachs, G., and Costa, O. da. Development of control laws for the simulation of a new transport aircraft. *Journal of Aerospace Engineering*, 223(2):141–156, 2009. doi:10.1243/09544100JAERO309.

[59] Hovakimyan, N., Lavretsky, E., and Sasane, A. Dynamic inversion for nonaffine-in-control systems via time-scale separation. *Journal of Guidance, Control and Dynamics*, 13(4):451–465, 2007. doi:10.1007/s10883-007-9029-1.

[60] Huang, C. S., Wang, S., and Chen, C. S. A radial basis collocation method for hamilton jacobi bellman equations. *Automatica*, 42(12):2201–2207, 2006. doi:10.1016/j.automatica.2006.07.01.

[61] Jategaonkar, R. V. *Flight Vehicle System Identification: A Time Domain Methodology*. AIAA Education Series, 2006.

[62] Jiao, L., Bo, L., and Wang, L. Fast sparse approximation for least squares support vector machine. *IEEE Trans. Neural Networks*, 18(3):685–697, 2007.

[63] Johnson, E. N. *Limited Authority Adaptive Flight Control*. PhD thesis, Geogia Institute of Technology, USA, Nov. 2000.

[64] Johnson, E. N. and Anthony, J. C. Pseudo-control hedging: A new method for adaptive control. In *Advances in Navigation Guidance and Control Technology Workshop*, Redstone Arsenal, Alabama, 2000. Nov.

[65] Karsmakers, P., Pelckmans, K., De Brabanter, K., Van hamme, H., and Suykens, J. A. K. Sparse conjugate directions pursuit with application to fixed-size kernel models. *Mach. Learn.*, 85(1):109–148, 2011.

[66] Keller, J. D., McKillip, R. M., and Kim, S. Aircraft flight envelope determination using upset detection and physical modeling methods. In *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, 2009. AIAA Paper: 2009-6256.

[67] Khalil, H. K. *Nonlinear Systems*. Prentice Hall, New Jersey, 2002.

[68] Klein, V. and Morelli, E. A. *Aircraft System Identification, Theory and Practice*. AIAA Education Series, 2006. pp.92-221.

[69] Klein, V., Batterson, J. G., and Murphy, P. C. Determination of airplane model structure from flight data by using modified stepwise regression. Nasa technical paper tp-1637, Langley Research Center, Washington, D.C., Oct. 1981.

[70] Krener, A. J. Approximate linearization by state feedback and coordinate change. *System and Control Letters*, 5:181–185, 1984.

[71] Krstic, M., Kanellakopoulos, I., and Kokotovic, P. V. *Nonlinear and Adaptive Control Design*. John Wiley and Sons, New Jersey, 1995.

[72] Kwatny, H. G., Bor-Chin Chang, G. B., Yasar, M., and Belcastro, C. Aircraft accident prevention: Loss-of-control analysis. In *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, 2009. AIAA Paper: 2009-6256.

[73] Lai, M. J. and Schumaker, L. L. *Spline Functions on Triangulations*. Cambridge, 2007. pp.18-27.

[74] Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Joran, M. I. Learning the kernel matrix with semidefinite programming. *Journal of machine learning research*, 5:27–72, 2004.

[75] Lavigne, L., Cazaurang, F., Fadiga, L., and Goupil, P. New sequential probability ratio test:validation on a380 flight data. *Control Engineering Practice*, 22:1–9, 2014. DOI:10.1016/j.conengprac.2013.09.008.

[76] Lawson, C. L. and Hanson, R. J. *Solving Least Squares Problems*. SIAM:Society for Industrial and Applied Mathematics, 1995. pp.148-168.

[77] Lee, S., Park, C., and Koo, J.-Y. Feature selection in the laplacian support vector machine. *Computational Statistics and Data Analysis*, 55(1):567–577, 2011.

[78] Levant, A. Higher-order sliding modes, differentiation and output-feedback control. *International Journal of Control*, 76(9):924–941, 2003. doi:10.1080/0020717031000099029.

[79] Li, Y., Goog, S., Sherrah, J., and Lddell, H. Support vector machine based multi-view face detection and recognition. *Image and Vision Computing*, 22(5):413–427, 2004.

[80] Liao, R. J., Zheng, H. B., Grzybowski, S., and Yang, L. J. Particle swarm optimization-least squares support vector regression based forecasting model on dissolved gases in oil-filled power transformers. *Electric Power Systems Research*, 81(12):2074–2080, 2011.

[81] Lin, C. S., Chiu, S. H., and Lin, T. Y. Empirical mode decomposition-based least squares support vector regression for foreign exchange rate forecasting. *Economic Modelling*, 29(6):2583–2590, 2012.

[82] Ljung, L. Recursive identification algorithms. *Circuits Systems Signal Processing*, 21:57–68, 2002.

[83] Ljung, L. and Soderstrom, T. *Theory and practice of recursive identification*. MIT Press, 1983. pp.67-135.

[84] Lombaerts, T. and Looye, G. H. N. Design and flight testing of nonlinear autoflight control laws. In *AIAA Guidance, Navigation and Control Conference*, Minneapolis, Minnesota, 2012. AIAA.

[85] Lombaerts, T., Schuet, S. R., Acosta, D. M., and Kaneshige, J. T. Safe maneuvering envelope estimation based on a physical approach. In *AIAA Guidance, Navigation and Control Conference*, Boston, MA, 2013. AIAA Paper:2013-4618.

[86] Lombaerts, T. J. *Fault Tolerant Flight Control*. PhD thesis, Delft University of Technology, The Netherlands, May 2010. pp.100-155.

[87] Lombaerts, T. J., Smaili, M. H., and Stroosma, O. Piloted simulator evaluation results of new fault-tolerant flight control algorithm. *Journal of Guidance, Control and Dynamics*, 32(6):1747–1765, 2009. doi:10.2514/1.44280.

[88] Lombaerts, T. J., Van Oort, E. R., and Chu, Q. P. Online aerodynamic model structure selection and parameter estimation for fault-tolerant control. *Journal of Guidance, Control and Dynamics*, 33(3):707–723, 2010. doi:10.2514/1.47256.

[89] Lombaerts, T. J., Chu, Q. P., Mulder, J. A., and Joosten, D. A. Modular flight control reconfiguration design and simulation. *Control Engineering Practice*, 19:540–554, 2011. doi:10.1016/j.conengprac.2010.12.008.

[90] Lombaerts, T. J. J., Chu, Q. P., and Mulder, J. A. *Automatic Flight Control System Design*. Lecture Notes AE4-301, Delft University of Technology, Faculty of Aerospace Engineering, 2005. Delft, The Netherlands.

[91] Looze, D. P., Weiss, J. L., Eternon, J. S., and Barett, N. M. An automatic redesign approach for restructurable control systems. *IEEE Control Systems Magazine*, 5: 16–22, 1985.

[92] Lygeros, J. On reachability and minimum cost optimal control. *Automatica*, 40(6): 917–927, 2004. doi:10.1016/j.automatica.2004.01.012.

[93] Maine, R. E. and Illif, K. W. *Identification of dynamic systems, applications to aircraft, part 1: The output error approach*. Agard flight test techniques series, volume 3, AGARDograph 300, 1986.

[94] Marzat, J., Piet-Lahanier, H., Damongeot, F., and Walter, E. Model-based fault diagnosis for aerospace systems: a survey. *Aerospace Engineering*, 00:1–32, 2012. DOI: 10.1177/0954410011421717.

[95] McEneaney, W. M. *Max-Plus Methods for Nonlinear Control and Estimation*. Birkhauser, Boston, 2005.

[96] Meyer, C. D. *Matrix Analysis and Applied Linear Algebra*. SIAM:Society for Industrial and Applied Mathematics, 2001. pp.403-450.

[97] Mitra, V., Wang, C.-J., and Banerjee, S. Text classification: a least square support vector machine approach. *Applied Soft computation*, 7(3):908–914, 2007.

[98] Mohanty, N. *Random Signals Estimation and Identification:Analysis and Applications*. Van Nostrand Reinhold, 1986. pp.266-291.

[99] Monaco, J. F., Ward, D. G., Barron, R. L., and Bird, R. A. Implementaion and flight test assesment of an adaptive and reconfigurable flight control system. In *AIAA Guidance, Navigation and Control Conference*, New Oleans, Lousiana, 1997. AIAA Paper 1997-3738.

[100] Morelli, E. A. Global nonlinear aerodynamic modeling using multivariate orthogonal functions. *Journal of Aircraft*, 32(2):270–277, 1995. doi:10.2514/3.46712.

[101] Morelli, E. A. Global nonlinear parametric modeling with application to f-16 aerodynamics. Nasa technical report acc-i-98010-2, Langley Research Center, Hampton, VA, Jan. 1998.

[102] Morelli, E. A. Real-time parameter estimation in the frequency domain. *Journal of Guidance, Control and Dynamics*, 23(5):812–818, 2000. doi:10.2514/2.4642.

[103] Morelli, E. A. Real-time aerodynamic parameter estimation without air flow angle measurements. In *AIAA Atomospheric Flight Mechanics Conference*, Toronto, Ontario Canada, 2010. AIAA.

[104] Morelli, E. A. and Cunningham, K. Aircraft dynamic modeling in turbulence. In *AIAA Atmospheric Flight Mechanics Conference*, Minneapolis, Minnesota, 2013. AIAA Paper: 2012-4650.

[105] Morelli, E. A. and Smith, M. Real-time dynamic modeling:data information requirements and flight test results. *Journal of Aircraft*, 46(6):1894–1905, 1999. doi:10.2514/1.40764.

[106] Mulder, J. A. Estimation of thrust and drag in nonsteady flight. In *Proceedings of the 4th IFAC Symposium, Identification and System Parameter Estimation*, Tbilisi, 1976.

[107] Mulder, J. A. Design and evaluation of dynamic flight test manoeuvers. Lr-497, Delft University of Technology, 1986.

[108] Mulder, J. A. *Design and evaluation of dynamic flight test manoeuvers*. PhD thesis, Delft University of Technology, The Netherlands, Oct. 1986.

[109] Mulder, J. A., Jonkers, H. L., Horsten, J. J., and Simons, J. L. *Analysis of aircraft performance, stability and control measurements*. AGARD Lecture Series, number 104, 1979.

[110] Mulder, J. A., Sridhar, J. K., and Breeman, J. H. *Identification of dynamic systems, applications to aircraft, part 2:nonlinear analysis and manoeuvre design*. Agard flight test techniques series, volume 3, AGARDograph 300, 1994.

[111] Mulder, J. A., Staveren, W. H. J. J. van, Vaart, J. C. van der, and Weerdt, E. de. *Flight Dynamics*. Lecture Notes AE3-302, Delft University of Technology, Faculty of Aerospace Engineering, 2006. Delft, The Netherlands.

[112] Narang-Siddarth, A., Peter, F., Holzapfel, F., and Valasek, J. Autopilot for a nonlinear non-minimum phase tail-controlled missile. In *AIAA Guidance, Navigation and Control Conference*, Maryland, 2014. AIAA Paper:2014-1293.

[113] Narendra, K. S. and Annaswamy, A. M. A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control*, 32(2):134–145, 1987.

[114] Nguyen, L. T., Ogburn, M. E., and Gilbert, W. P. Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability. Nasa technical paper 1538, Langley Research Center, Hampion, Virginia, Dec. 1979.

[115] Ning, H. W., Jing, X. J., and Cheng, L. Online identification of nonlinear spatiotemporal systems using kernel learning approach. *IEEE Transactions on neural networks*, 22(9):1381–1394, 2011.

[116] NLR. Civil aviation safety data 1993-2007. Technical report, Civil Aviation Authority of the Netherlands, Netherlands, 2008.

[117] Ong, C. S., Smola, A. J., and Williamson, R. C. Learning the kernel with hyperkernels. *Journal of machine learning research*, 6:1043–1071, 2005.

[118] Oppenheimer, M. and Doman, D. A method for including control effector interactions in the control allocation problem. In *AIAA Guidance, Navigation and Control Conference*, Hilton Head, SC, USA, 2007. AIAA :2007-6418.

[119] Patton, R. Fault tolerant control systems: The 1997 situation. In *Proceedings of IFAC Symposium on SAFEPROCESS*, pages 1033–1054, Kingston Upon Hull, England, U.K., 1997. International Federation of Automatic Control.

[120] Pillonetto, G., Quang, M. H., and Chiuso, A. A new kernel-based approach for nonlinear system identification. *IEEE Trans. on Automatic Control*, 56(12):2825–2840, 2011.

[121] Polat, K. and Gunes, S. Breast cancer diagnosis using least square support vector machine. *Digital Signal Processing*, 17(4):694–701, 2007.

[122] Powell, W. B. *Approxiate Dynamic Programming: solving the curses of dimensionality*. Wiley-Intersciense, Nashua, New Hampshire, 2007.

[123] Qiu, G., Wu, Z., and Du, Q. Application of least square support vector machine for thunderstorm prediction. In *Proceeding of the Eighth World Congress on Intelligent Control and Automation*, pages 345–349, Washington, DC, 2010. IEEEXplore.

[124] Quan, T., Liu, X., and Liu, Q. Weighted least squares support vector machine local region method for nonlinear time series prediction. *Applied Soft computation*, 10 (2):562–566, 2010.

[125] Raimund, S. The upper bound theorem for polytopes: an easy proof of its asymptotic version. *Computational Geometry*, 5:115–116, 1995. doi:10.1016/0925-7721(95)00013-Y.

[126] Scholkopf, B. and Smola, A. J. *Learning with Kernels*. Cambridge:MIT Press, 2002.

[127] Schuet, S., Lombaerts, T., Acosta, D., Wheeler, K., and Kaneshige, J. An adaptive nonlinear aircraft maneuvering envelope estimation approach for online applications. In *AIAA Guidance, Navigation and Control Conference*, Maryland, 2014. AIAA Paper:2014-0268.

[128] Shewchuk, J. R. Delauney refinement algorithms for triangular mesh generation. *Computational Geometry*, 22:21–74, 2001. doi:10.1016/S0925-7721(01)00047-5.

[129] Sieberling, S. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *Journal of Guidance, Control and Dynamics*, 33(6):1732–1742, 2010. doi:10.2514/1.49978.

[130] Slotine, J.-J. E. and Li, W. *Applied Nonlinear Control*. Prentice Hall, New Jersey, 1991.

[131] Smaili, M., Breeman, J., Lombaerts, T., and Joosten, D. A simulation benchmark for integrated fault tolerant flight control evaluation. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Keystone, Colorado, 2006. AIAA Paper 2006-6471.

[132] Smaili, M., Breeman, J., and Lombaerts, T. A simulation benchmark for aircraft survivability assessment. In *26th International Congress of Aeronautical Sciences*, pages 1–12, Anchorage, Alaska, 2008. The International Council of the Aeronautical Science, ICAS 2008-9.3.2.

[133] Soderstrom, T., Ljung, L., and Gustavsson, I. A theoretical analysis of recursive identification methods. *Automatica Journal*, 14:231–244, 1987.

[134] Song, Y., Campa, G., Napolitano, M. R., Seanor, B., and Perhinsch, M. Online parameter estimation techniques comparison within a fault tolerant flight control system. *Journal of Guidance, Control and Dynamics*, 25(3):528–537, 2002. doi:10.2514/2.4913.

[135] Sonneveldt, L., Van Oort, E. R., Chu, Q. P., and Mulder, J. A. Nonlinear adaptive trajectory control applied to an f-16 model. *Journal of Guidance, Control and Dynamics*, 32(1):25–39, 2009. doi:10.2514/1.38785.

[136] Sonneveldt, L., Oort, E. R. van, Chu, Q. P., and Mulder, J. A. Immersion and invariance based nonlinear adaptive flight control. In *AIAA Guidance, Navigation and Control Conference*, Toronto, Ontario Canada, 2010. AIAA:2010-7690.

[137] Stark, J. Adaptive model selection using orthogonal least squares methods. In *Mathematical, Physical and Engineering Sciences*, pages 21–42, Britain, 1997. The Royal Society.

[138] Sun, L. G., Visser, C. C. de, and Chu, Q. A new substitution based recursive b-splines method for aerodynamic model identification. In *The CEAS EuroGNC 2013*, Delft, Netherlands, 2013. Springer.

[139] Sun, L. G., Visser, C. C. de, Chu, Q. P., and Falkena, W. Hybrid sensor-based backstepping control approach with its application to fault-tolerant flight control. *Journal of Guidance, Control and Dynamics*, 37(1):59–71, 2013. doi:10.2514/1.61890.

[140] Sun, L. G., Visser, C. C. de, Chu, Q. P., and Falkena, W. A joint sensor based backstepping approach for fault-tolerant flight control of a large civil aircraft. In *AIAA Guidance, Navigation and Control Conference*, Boston, MA, 2013. AIAA.

[141] Sun, L. G., Visser, C. C. de, Chu, Q. P., and Mulder, J. A. Online aerodynamic model identification using a recursive sequential method for multivariate splines. *Journal of Guidance, Control and Dynamics*, 36(5):1278–1288, 2013. doi:10.2514/1.60375.

[142] Sun, L. G., Visser, C. C. de, and Chu, Q. A new multivariate tensor-product simplex b-splines with its application in real-time modular adaptive flight control. In *AIAA Guidance, Navigation and Control Conference*, Maryland, 2014. AIAA Paper:2014-0087.

[143] Sun, L. G., Visser, C. C. de, Chu, Q. P., and Mulder, J. A. Joint sensor based backstepping for fault-tolerant flight control. *Journal of Guidance, Control and Dynamics, Accepted*, 00(0):xx–xx, 2014. doi:10.2514/1.61890.

[144] Sun, L. G., Visser, C. C. de, Chu, Q. P., and Mulder, J. A. A novel adaptive kernel method with kernel centers determined by a support vector regression approach. *Neurocomputing*, 124(1):111–119, 2014. doi:10.1016/j.neucom.2013.07.023.

[145] Suykens, J. A. K. and Vandewalle, J. Least squares support vector machine classifiers. *Neural process, Lett.*, 9(3):293–300, 1999.

[146] Suykens, J. A. K., Lucas, L., and Vandewalle, J. Sparse approximation using least squares vector machines. In *IEEE International Symposium on Circuits and Systems*, pages 757–760, Geneva, 2000. IEEE.

[147] Suykens, J. A. K., De Brabanter, J., Lucas, L., and Vandewalle, J. Weighted least squares support vector machines:robustness and sparse approximation. *Neuralcomputing*, 48(1):85–105, 2002.

[148] Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. *Least squares support vector machines*. World Scientific, 2002.

[149] Tang, L., Roemer, M., Bharadwaj, S., and Belcastro, C. An integrated health assessment and fault contingency management system for aircraft. In *AIAA Guidance, Navigation and Control Conference*, Honolulu, Hawaii, 2008. AIAA Paper: 2008-6505.

[150] Tang, L., Roemer, M., Ge, G., Crassidis, A., and Belcastro, C. Methodologies for adaptive flight envelope estimation and protection. In *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, 2009. AIAA Paper: 2009-6260.

[151] Tekles, N., Xargay, E., Choe, R., Hovakimyan, N., Gregory, I. M., and Holzapfel, F. Physically integrated reference model and its aids in validation of requirements to flight control systems. In *AIAA Guidance, Navigation and Control Conference*, Maryland, 2014. AIAA Paper:2014-0269.

[152] The UK Civil Aviation Authority, . *Global Fatal Accident Review:2002-2011*. The Stationery Office, 2013. CAP 1036.

[153] Tikhonov, A. N. Systems of differential equations containing small parameters in the derivatives. *Mathematical Sb. (N.S.)*, 31(3):575–586, 1952. in Russian.

[154] Tischer, M. B. and Remple, R. K. *Aircraft and Rotorcraft System Identification, Engineering Methods with Flight Test Examples*. AIAA Education Series, 2006. pp.1-20.

[155] Tol, H. J., Visser, C. C. de, Kampen, E. van, and Chu, Q. P. Multivariate simplex spline based nonlinear dynamic inversion control of high performance aircraft. In *AIAA Guidance, Navigation and Control Conference*, Boston, MA, 2013. AIAA Paper:2013-4925.

[156] Tol, H. J., Visser, C. C. de, Sun, L. G., Kampen, E. van, and Chu, Q. P. Nonlinear multivariate spline based adaptive control allocation with online aerodynamic model identification. *Journal of Guidance, Control and Dynamics, Under Review*, 38(2):606–610, 2014.

[157] Tol, H. J., Visser, C. C. de, Kampen, E. van, and Chu, Q. P. Nonlinear multivariate spline-based control allocation for high-performance aircraft. *Journal of Guidance, Control and Dynamics*, 36(2):606–610, 2014. doi:10.2514/1.G000065.

[158] Vahram, S., Kalmanje, K., and Nhan, N. Adaptive control of a transport aircraft using differential thrust. Nasa technical paper 1108, NASA Ames Research Center, Moffett Field, Aug. 2009.

[159] Eykeren, L. van and Chu, Q. P. Sensor fault detection and isolation using adaptive extended kalman filter. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process*, Mexico city, Mexico, 2012. IFAC:pp.1155-1160.

[160] Eykeren, L. van and Chu, Q. P. Sensor fault detection and isolation for aircraft control systems by kinematic relations. *Control Engineering Practice*, 00:xxx–xxx, 2014. doi:10.1016/j.conengprac.2014.02.017.

[161] Eykeren, L. van and Chu, Q. P. Sensor fault detection and isolation for redundant air data sensors. In *AIAA Guidance, Navigation and Control Conference*, Maryland, 2014. AIAA Paper:2014-1147.

[162] Oort, E. van. *Adaptive Backstepping Control and Safety Analysis for Modern Fighter Aircraft*. PhD thesis, Delft University of Technology, The Netherlands, Apr. 2011.

[163] Oort, E. R. van, Sonneveldt, L., Chu, Q. P., and Mulder, J. A. Full-envelope modular adaptive control of a fighter aircraft using orthogonal least squares. *Journal of Guidance, Control and Dynamics*, 33(5):1461–1472, 2010. doi:10.2514/1.48175.

[164] Oort, E. R. van, Chu, Q., and A., M. J. Maneuver envelope determination through reachability analysis. In *The CEAS EuroGNC 2011*, Munich, Germany, 2011. Springer.

[165] Vapnik, V. N. *The Nature of Statistical Theory*. New York:Springer-Verlag, 1995.

[166] Verhaegen, M. and Verdult, V. *Filtering and System Identification, A Least Squares Approach*. Cambridge University Press, UK, 2007.

[167] Vincent, A. A. and Hassapis, G. D. Nonlinear model identification and adaptive model predictive control using neural networks. *ISA Transactions*, 50(1):177–194, 2011.

[168] Voth, C. and Ly, U.-L. Total energy control system autopilot design with constrained parameter optimization. In *American Control Conference*, pages 1332–1337, San Diego, California, 1990. IEEE.

[169] Yu, Z., Wong, H.-S., and Wen, G. A modified support vector machine and its application to image segmentation. *Image and Vision Computing*, 29(1):29–40, 2011.

[170] Zeng, X. Y. and Chen, X. W. Smo-based pruning methods forsparse least squares support vector machines. *IEEE Trans. Neural Networks*, 16(6):1541–1546, 2005.

[171] Zhang, F. B., Braun, S., and Holzapfel, F. Physically integrated reference model and its aids in validation of requirements to flight control systems. In *AIAA Guidance, Navigation and Control Conference*, Maryland, 2014. AIAA Paper:2014-0962.

[172] Zhang, X. D. *Matrix Analysis and Applications*. Tsinghua University Press, 2004. pp.659-662.

[173] Zhang, Y. Fault tolerant control systems:historical review and current research. technical paper, Centre de Recherche en Automatique de Nancy (CRAN), Universite Henri Poincare, Nancy, France, 2005.

[174] Zhang, Y. and Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229–252, 2008. doi:10.1016/j.arcontrol.2008.03.008.

[175] Zhang, Y. M. and Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32:229–252, 2008.

[176] Zhao, Y. P. and Sun, J. G. Recursive reduced least squares support vector regression. *Pattern Recognition*, 42(5):837–842, 2009. doi:10.1016/j.patcog.2008.09.028.

[177] Zhao, Y. P. and Sun, J. G. Fast online approximation for hard support vector regression and its application to analytical redundancy for aeroengines. *Chinese Journal of Aeronautics*, 23:145–152, 2010.

[178] Zhao, Y. P. and Sun, J. G. Multikernel semiparametric linear programming support vector regression. *Expert Systems with Applications*, 38:1611–1618, 2011.

[179] Zhao, Y. P., Sun, J. G., Du, Z. H., and Zhang, Z. A. An improved recursive reduced least squares support vector regression. *Neurocomputing*, 87(1):1–9, 2012. doi:10.1016/j.neucom.2012.01.015.

[180] Zhu, Y. and Li, X. R. Recursive least squares with linear constraints. *Communications in Information and Systems*, 7(3):287–312, 2007.

[181] Zolghadri, A. Advanced model-based fdir techniques for aerospace systems: Today challenges and opportunities. *Progress in Aerospace Sciences*, 53:18–29, 2012. DOI:10.1016/j.paerosci.2012.02.004.

[182] Zolghadri, A., Castang, F., and Henry, D. Design of robust fault detection filters for multivariable feedback systems. *International Journal of Modelling and Simulation*, 26:17–26, 2006.

[183] Zolghadri, A., Henry, D., Cieslak, J., Efimov, D., and Goupil, P. *Fault Diagnosis and Fault-Tolerant Control and Guidance for Aerospace Vehicles, from theory to application*. Springer, Series: Advances in Industrial Control, London, UK, 2014.

SAMENVATTING

Model- en sensorgebaseerde niet-lineaire adaptieve vliegtuigbesturing met online system identificatie

Li Guo Sun

Er is een consensus dat veel accidenten waarbij verlies van controle (LOC) optreedt, veroorzaakt door ernstige schade aan het vliegtuig of door het falen van een component, vermeden zouden kunnen worden omdat de vluchtperformantie hersteld kan worden door de overblijvende regelmogelijkheden te gebruiken. Dit niettegenstaande dat de veilige manoeuvreerbaarheid van het vliegtuig na het accident onoverkomelijk gereduceerd zal zijn. Onconventionele regelstrategieën, gebaseerd op moderne regeltechnieken en een uitgebreide rekenkracht, zijn essentieel om het meeste te halen uit de gereduceerde, overblijvende regelmogelijkheden na een defect en om de vliegperformantie te herstellen of een veilige landing uit te kunnen voeren. Één zo'n onconventionele regeltechniek is het actieve fout-tolerante vliegtuigregelsysteem (FTFC), ontworpen om veranderingen in de vliegtuigdynamica ten gevolge van structurele, actuator of sensor fouten te ontdekken en om via een adaptief reconfiguratiemechanisme zich aan te passen aan de schade of fout. De actieve FTFC kan overweg met onvoorziene en meerdere simultane defecten.

De gehele systeemarchitectuur van een actieve FTFC zou in een ideaal geval moeten bestaan uit een Fout Detectie en Diagnose (FDD) module, een onderdeel voor de reconstructie van de toestand van het systeem, een component met een reconfigureerbare regelaar, een controle toewijzingseenheid en een component voor de beveiliging van het vliegdomein. FTFC systemen kunnen normaal gezien ingedeeld worden in twee verschillende types: modelgebaseerde systemen en modelvrije systemen, afhankelijk van het feit of één van de componenten een aerodynamisch model van het vliegtuig gebruikt of niet. Één van de onderdelen van een modelgebaseerde FTFC is de aerodynamisch model identificatie (AMI) module, die een nauwkeurig vliegtuigmodel beschikbaar stelt aan een indirect adaptieve niet-lineaire regelaar in het onderdeel met de reconfigureerbare regelaar, aan een dynamisch algoritme voor de definiëring van het vliegdomein, of aan een FDD module. Een AMI methode met een fysische, interpreteerbare modelstructuur kan structurele defecten of een defect van één van de stuurvlakken detecteren en zelfs kwantificeren door veranderingen in de stabiliteits- en regelafgeleiden te observeren.

Er bestaan veel mogelijke regelstrategieën die over reconfiguratie mogelijkheden beschikken. Deze reconfigureerbare regelstrategieën zijn gebaseerd op een wijd scala van reconfiguratiemechanismen zoals *switching, model following, matching* en *adaptieve compensatie*. Één van deze methodes is niet-lineaire adaptieve regelaars, die

reconfiguratie behalen door middel van compensatie. Deze methode trekt steeds meer aandacht in het specifieke onderzoeksgebied dat zich toelegt op de besturing van luchtvaart- en ruimtevaarttuigen. Niet-lineaire adaptieve regelmethodes kunnen onderverdeeld worden in *directe* adaptieve regelaars en *indirecte* adaptieve regelaars, waarbij het verschil erin bestaat dat de laatste een *online* systeemmodel gebruikt. Indirecte adaptieve regelaars worden ook modelgebaseerde of modulaire adaptieve regelaars genoemd, en hebben verschillende voordelen ten opzichte van de directe adaptieve regelaars en andere modelvrije regelstrategieën. Één van deze voordelen is dat een modulaire regelstrategie potentieel meer efficiënt is en minder stuuracties nodig heeft. Zo'n efficiënte controletechniek kan gecreëerd worden door nuttige dempingtermen van een geïdentificeerd model te behouden in de *closed-loop*. Dit wordt toegeschreven aan de goede eigenschappen van vele ontwerptechnieken voor regelmethodes zoals *backstepping*, zodat er kan gekozen worden om de dynamica van het originele systeem te compenseren of te behouden tijdens het ontwerpproces. Modulaire adaptieve besturing heeft ook een inherent nadeel, enkel de *input-to-state* stabiliteit kan gegarandeerd worden, i.e., modulaire adaptieve besturing kan niet de stabiliteit van het gehele *closed-loop* systeem garanderen omdat het stabiliteitsbewijs gebaseerd is op het *certainty equivalence* principe. De zwakheid van het *certainty equivalence* principe, i.e., een convergentieprobleem van de modelparameters, kan gecompenseerd worden door de modelnauwkeurigheid of betrouwbaarheid te verbeteren. Om dit te bereiken is het essentieel om geavanceerde, krachtige identificatiemethodes te ontwikkelen voor het identificeren van aerodynamische modellen die capabel zijn om veranderingen in de vliegdynamica te ontdekken, zowel tijdens sterke manoeuvres als tijdens een post-faling conditie.

De beveiliging van het vliegdomein is een noodzakelijke techniek dat ontwerpers van regelaars zouden moeten toepassen om LOC incidenten te vermijden, daarbij rekening houdend met sterke manoeuvres en/of zwaar verstoorde vliegcondities door de aanhoudende storing. Een component voor de beveiling van het vliegdomein zou een piloot moeten voorzien van een indicatie voor het veilige vliegdomein en limieten zetten op de referentiesignalen voor de interne regelaar zodat de stuurdoelen bereikt kunnen worden.

Een aerodynamisch vliegtuigmodel dat geldig is over het gehele vliegdomein speelt een cruciale rol in een modulaire adaptieve regeltechniek en vliegdomeinbeveiliging. Een globaal geldig model is nodig voor een modulaire adaptieve regeltechniek zodat de ontworpen regelaar optimaal kan functioneren over het gehele operationele bereik van het vliegtuig. Eenmaal geschat kan het globale model van een modelgebaseerde adaptieve regeltechniek opgeslagen worden voor later gebruik, wanneer dezelfde vluchtconditie opnieuw wordt tegengekomen. Essentieel voor een modelgebaseerde regeltechniek is een nauwkeurig aerodynamisch vliegtuigmodel, dat bovendien ook nodig is voor de vliegdomeinbeveiliging. Het spreekt voor zich dat het geschatte aerodynamisch model geldig moet zijn over het gehele vliegdomein voor een specifieke configuratie zodat een evolutiealgoritme de grenzen van het veilige vliegdomein kan schatten voor de geldende vliegcondities. Echter, er zijn maar enkele modelidentificatiemethodes die in staat zijn om een globaal geldig aerodynamisch model te schatten, en elke mogelijke methode heeft verschillende tekortkomingen of limitaties die het toepassen

van deze methodes voor het schatten van een vliegtuigmodel bemoeilijken. Bijvoorbeeld, neurale netwerken resulteren gebruikelijk in een niet-transparante modelstructuur die moeilijk te interpreteren is door gebruik te maken van de fysische kennis van het systeem en bovendien kunnen er convergentieproblemen optreden. De meeste kernfunctiemethodes behoren tot de categorie van niet-parametrische methodes, die van nature even veel kernfuncties nodig hebben als er data punten zijn. Men moet er rekening met houden dat enkel modelidentificatiemethods van het *equation-error* type onderzocht zijn geweest in dit werk, aangezien de aanname werd gemaakt dat voldoende nauwkeurige schattingen van de vliegtuigtoestand beschikbaar waren.

Een alternatieve methode voor de modulaire adaptieve reconfigureerbare regeltechniek is de incrementele niet-lineaire besturing gebaseerd op de acceleratiemetingen (AMINC). Een nauwkeurige schatting van het vliegtuigmodel tijdens sterke manoeuvres of tijdens een transitiemoment na het falen van het vliegtuig is moeilijk te behalen. Incrementele regeltechnieken zoals incrementele niet-lineaire dynamische inversie (INDI), incrementele *backstepping* (IBKS) en sensor-gebaseerde *backstepping* (SBB) zijn geschikt voor reconfigureerbare ontwerpen voor vliegregelaars in die zin dat deze technieken geen volledig vliegtuigmodel nodig hebben.

De hoofdonderzoeksvraag voor dit werk is: op welke manier kan een geavanceerd faut-tolerant regelsysteem ontworpen worden zodat de overlevingskansen van een vliegtuig verhoogd kunnen worden? Deze vraag leidde tot twee subvragen:

- Hoe kunnen de kandidaatmethodes voor functieapproximatie, i.e., multivariate simplex B-splines en kernfunctie methodes, verbeterd worden op vlak van benaderingsnauwkeurigheid en rekenkundige efficiëntie, zodat tegemoetgekomen kan worden aan de voorwaarden voor modelgebaseerde adaptieve besturing en *online* vliegdomeinbeveiliging?

- Wat zijn de voordelen van gebruik te maken van een regeltechniek gebaseerd op versnellingsmetingen, i.e., de sensor-gebaseerde *backstepping*, als een alternatief voor een modelgebaseerde adaptieve besturingsmethode, wanneer een reconfigureerbare regelaar dient ontworpen te worden die kan omgaan met schade?

In het geval van reconfigureerbare besturing zou het geïdentificeerde model de regelaar in de mogelijkheid moeten stellen om actieve reconfiguratie te behalen en de besturingsperformantie te herstellen. Om deze vragen te beantwoorden zijn vier verschillende globale identificatiemethodes en twee verschillende niet-lineaire incrementele adaptieve regelaars ontworpen.

Twee modelidentificatiemethodes maken gebruik van een parametrische modelstructuur, zijnde standaard multivariate simplex B-splines. De focus tijdens het onderzoeksproces voor deze methods lag voornamelijk op hoe een snelle parameteridentificatie bereikt kan worden. In de derde identificatiemethode werd een nieuwe modelstructuur, genaamd tensor-product simplex B-splines, uitgebreid van een ééndimensionale toepassing naar een multidimensionale toepassing, waarbij de focus lag op het demonstreren van het voordeel van deze nieuwe samengestelde modelstructuur in termen van de flexibiliteit bij de selectie van de modelstructuur, de rekenefficiëntie en de benaderingsnauwkeurigheid. De vierde methode maakt gebruik van een kernfunctie modelstructuur die ook parametrisch is. De nieuwe recursieve kernfunctiemethode is

ontwikkeld door een klassieke recursieve kernfunctiemethode te combineren met een nieuwe *support vector* regressie methode.

Een modelidentificatiemethode die gebruik maakt van standaard multivariabele simplex B-splines heeft veel voordelen, zoals het voorkomen van *over-fitting* dat voorkomt bij gewone polynome methodes die gebruik maken van een triangulatietechniek. Het benaderingsvermogen van een methode gebaseerd op simplex B-splines is gedefiniëerd door de per-simplex polynoomorde en de orde van de continuïteit, en kan verhoogd worden door het laten toenemen van het aantal subdomeinen in de triangulatie. Deze benaderingsmethode gebaseerd op simplex B-splines garandeert dat de output begrensd is door de maximum en minimum B-coëfficiënten, wat de certificatie voor toekomstige toepassingen faciliteert. De lineaire regressieve formulatie van de methode gebaseerd op B-splines laat toe om de meeste recursieve parameterschattingsmethodes toe te passen. Bovendien heeft de op de simplex B-spline gebaseerde methode een *sparse* eigenschap, die de rekenkundige efficiëntie kan verhogen door gebruik te maken van gedistribueerde berekeningen of andere moderne rekenkundige technieken. Echter, in het geval dat de functiedimensie 4 of meer is, kan de toepassing van een simplex B-spline methode leiden tot een snelle toename aan het aantal onbekenden, wat opnieuw resulteert in een hoge rekenkundige last, gerelateerd aan de vereisten voor continuïteit en het bijwerken van de covariantiematrix.

Om de rekenkundige efficiëntie van de modelidentificatiemethodes die gebruik maken van simplex B-splines te verhogen, zijn er twee recursieve lineaire-regressie modelidentificatiemethodes ontwikkeld in dit proefschrift: een op substitutie gebaseerde multivariabele simplex B-spline methode (SB-MVSB) en een recursieve sequentiële multivariabele simplex B-spline (RS-MVSB) methode. In de SB-MVSB methode is een efficiënte, recursieve solver ontwikkeld voor een gelimiteerd lineair regressieprobleem wanneer simplex B-splines gebruikt worden. Het gelimiteerde lineaire regressieprobleem wordt omgezet in een ongelimiteerd lineair regressieprobleem door gebruik te maken van een algemene oplossing voor *equality constraints*. Er werd aangetoond dat deze transformatie de schaal van het identificatieprobleem in termen van het aantal onbekenden kan reduceren, en dus op deze manier de benodige rekenkundige kracht voor de modelidentificatiemethode kan reduceren.

De RS-MVSB methode bestaat uit twee opeenvolgende procedures, uitgevoerd bij elke stap in het algoritme. De eerste procedure vernieuwt een lokaal model in plaats van een globaal model. De noodzakelijkheid van het vernieuwen van de complete covariantiematrix wordt vermeden door enkel één lokaal model te vernieuwen, en op deze manier wordt de rekenkundige efficiëntie van deze methode sterk verbeterd. De tweede procedure garandeert een continue transitie tussen dit lokaal model en de omringende lokale modellen.

De rekenkundige complexiteit van de SB-MVSB en de RS-MVSB methodes werd gepresenteerd vanuit een mathematisch oogpunt. Daarna werden ze gevalideerd d.m.v. gesimuleerde testdata gegenereerd met een F-16 model met een hoge waarheidsgetrouwheid. De simulatieresultaten demonsteerden dat beide methodes een veel hogere benaderingsnauwkeurigheid kunnen bereiken dan gewone methodes gebaseerd op polynomen, en dat beide vele malen sneller, e.g., 10, kunnen zijn dan equality constraint recursive least squares gebaseerde MVSB (ECRLS-MVSB) methodes. Deze

tweede eigenschap maakt het *onboard* gebruik van deze methodes mogelijk.

Tensor product simplex (TPS) B-splines hebben een samengestelde structuur die meer flexibiliteit voorziet dan een standaard simplex B-spline model tijdens het selecteren van de modelstructuur. Door gebruik te maken van TPS B-splines kunnen inputs van verschillende dimensies op verschillende manieren behandeld worden, afhankelijk van hun karakteristieken gekend van *a priori* kennis. In dit proefschrift werd het concept van TPS B-splines uitgebreid van ééndimensionele toepassingen naar een meer algemeen concept voor multidimensionele toepassingen. Door het verminderen van het aantal basispolynomen in de regressievector hebben TPS B-splines het potentieel om minder rekenkundige kracht nodig te hebben in vergelijking met standaard B-splines. De TPS B-spline methode werd gevalideerd met behulp van een dataset gegenereerd met een F-16 model met een hoge waarheidsgetrouwheid. De simulatieresultaten toonden aan dat TPS B-splines een hogere benaderingsnauwkeurigheid kunnen bereiken met minder B-coëfficienten dan standaard B-splines.

Voor de toepassing van aerodynamische modelidentificatie werden er twee gelijkaardige recursieve parametrische kernfunctie methodes ontwikkeld, genaamd "weight varying least squares support vector regression (WV-LSSVR)" en "Gaussian process kernel based LSSVR (GPK-LSSVR)". De focus van dit werk was het verbeteren van de benaderingsnauwkeurigheid van een recursieve kernfunctie methode door het kiezen van een optimale set van kernfuncties voor het kernfunctieschema. Een offline methode, genaamd "improved recursive reduced LSSVR (IRR-LSSVR)", werd toegepast om de optimale kernfuncties voor een klassieke recursieve kernfunctiemethode te bepalen. De nieuwe kernfunctiemethode werd gevalideerd op een serie van publiek beschikbare referentiedata die bekend is bij onderzoekers in het domein van patroonherkenning. GPK-LSSVR heeft een hogere benaderingsnauwkeurigheid dan WV-LSSVR, en beide demonstreerden een betere benaderingsnauwkeurigheid dan klassieke recursieve kernfunctiemethodes gebaseerd op "k-means clustering".

Met het doel om een reconfigureerbare regelaar te voorzien van een krachtige, onconventionele regeltechniek die zich kan aanpassen aan het plotse structurele of actuator falen, werd een nieuw type incrementele regelaar gebaseerd op versnellingsmetingen onderzocht. De geprefereerde, modelvrije, incrementele regeltechniek in dit proefschrift was de SBB techniek, die initieel ontwikkeld werd voor regelaars voor niet-lineaire, nonaffine-in-control systemen. De SBB techniek heeft accurate prestaties voor het volgen van een referentiesignaal door middel van benaderende dynamische inversie. De SBB techniek werd uitgebreid om te kunnen omgaan met plotse veranderingen in het vliegtuimodel door het structureel falen of door actuator fouten. Een hybride, dubbele lus, hoekregelaar en een verbonden dubbele lus hoekregelaar werden ontworpen voor het RECOVER model. In de hybride, dubbele lus hoekregelaar werd de hoekregellus ontworpen door middel van een niet-lineaire dynamische inversie regelwet, en de hoeksnelheidregelaar door middel van de SBB techniek. In de verbonden, dubbele lus hoekregelaar werd de gehele regelaar ontworpen via de backstepping techniek, waarbij elke lus recursief gestabiliseerd werd. Beide hoekregelaars werden gevalideerd door gebruik te maken van het RECOVER model, met de focus op situaties waarbij de vliegperformantie verstoord word door het structureel falen of falen van componenten. Twee referentiele foutscenarios werden geselecteerd: een runaway van het

richtingsroer en het scenario van El Al vlucht 1862 met de afscheuring van twee motoren. De simulatieresultaten toonden aan dat beide regeltechnieken in staat zijn om de veiligheid na het falen van het vliegtuig te garanderen en bovendien capabel zijn om het referentiesignaal te volgen. In vergelijking met de hybride NDI/SBB hoekregelaar had de verbonden SBB hoekregelaar een betere performantie bij het volgen van het referentiesignaal voor de zijwaartse slipbeweging, in het bijzonder in het scenario waarbij de motoren verloren werden.

Een SBB regelaar omvat een tijdschaal parameter, andere incrementele regeltechnieken zoals incrementele NDI (INDI) en incrementele backstepping (IBKS) maken gebruik van een regeldoeltreffendheidsmatrix. Voordat onderzocht kan worden hoe deze parameters de performantie van een incrementele regelaar beïnvloeden, moeten de parametervariaties van de regeldoeltreffendheidsmatrix geschat en geanalyseerd worden. De TPS B-spline methode en een immersie en invariante (I&I) methode werden gekozen om de regeldoeltreffendheidsmatrix voor een F-16 vliegtuig te schatten. Hoewel de I&I methode initieel niet gericht was op hoge modelaccuraatheid, werd er in dit proefschrift aangenomen dat de methode variabele trends van de regelafgeleiden kon schatten. De simulatieresultaten toonden aan dat TPS B-splines de veranderingen van de regelafgeleiden beter kon vastleggen in termen van consistentie dan de I&I methode. In het geval van de F-16 is het niet duidelijk dat de regeldoeltreffendheidsmatrix de performantie van incrementele regelaars beïnvloed tijdens een gematigd vliegmanoeuvre in termen van de aanvlieghoek en snelheid.

Verder onderzoek op het onderwerp van modulaire, adaptieve, reconfigureerbare regelaars is nodig. Een voorbeeld is het inbrengen van de SB-MVSB of de WV-LSSVR methodes in regeltechnieken om te controleren in hoever deze methodes gepast zijn voor modulaire, adaptieve regelaars in termen van benaderingsnauwkeurigheid en onboard rekenkundige efficiëntie. Verder onderzoek naar reconfigureerbare regeltechnieken gebaseerd op versnellingsmetingen zou ook gebruik moeten maken van evaluaties met de SIMONA onderzoekssimulator, realistische testvluchten met UAV's en onderzoeksvliegtuigen.

CURRICULUM VITAE

Li Guo SUN was born in Tangshan, People's Republic of China on March 07, 1984.

In 2004, he enrolled as a student at Nanjing University of Aeronautics and Astronautics (NUAA), China. He received a B.Sc with distinction from NUAA in 2008 with a major in Aircraft Power and Engineering, Department of Energy and Power Engineering. His graduation thesis was awarded a prize. In parallel, he started his studies for his master degree in November of 2006 (1.5 years earlier than his classmates) having been exempted from the entrance examination. In his master stage, he joined the Control and Simulation group, division of Aerospace Propulsion Theory and Engineering, department of Energy and Power Engineering, NUAA. As a master student, he participated two projects, both of which were funded by the Chinese aerospace research institutions, between 2008 and 2010. One of the projects was focused on modeling, control and optimization of an aero-engine, the other was focused on integrated flight/propulsion control (IF/PC). Working on IF/PC for his master thesis, he obtained his M.Sc. degree from NUAA in June of 2010. Li Guo did an internship in the MTU Aero-engine Maintenance Company, Zhuhai, China between July 28 and August 25 in 2007.

Since September of 2010, he has worked as a Ph.D. candidate at the division of Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands. His duties include: grading examinations; supervising and mentoring master students and assessing their work. As described in this thesis, Li Guo's research is mainly focused on nonlinear system (aerodynamic model) identification, multivariate spline theory and fault-tolerant flight control. His research interests also include active disturbance rejection control, robust control and integrated flight/propulsion control. Additionally, he has a solid background in flight dynamics, fluid mechanics, thermodynamics, heat transfer theory, combustion theory, turbine theory, propulsion theory and material mechanics.

Li Guo designed and built a small fixed-wing model aeroplane, with a wing span of 1.55 meters, during his PhD as a hobby. The UAV was built, in cooperation with 9 other PhD students, from basic raw materials up. Each component was designed, ordered and modified by the team members. It was successfully flown and tested in July of 2013.

Publications During My Master Stage
Master thesis with distinction: Performance Seeking Control of Aero-Engine Based on Integrated Helicopter/Turboshaft Engine Simulation, June, 2010

1. **Sun Liguo**, Sun Jianguo, Zhang Haibo, Research on augmented LQR controller design based on nonlinear integrated helicopter/engine model, Journal of Aerospace Power, 2010, 25(2), written in Chinese and an English abstract available online

2. **Sun Liguo**, Sun Jianguo, Zhang Haibo, Torque feed-forward control of engine/helicopter system based on support vector regression, Journal of Aerospace Power, 2011, 26 (3), written in Chinese and an English abstract available online

3. **Sun Liguo**, Wang Rixian, Sun Jianguo, Subsystem disturbance rejection control of turbo-shaft engine/helicopter based on cascaded ADRC, Journal of Beijing University of Aeronautics and Astronautics, 2011, vol. 37, written in Chinese and an English abstract available online

4. Zhang Haibo, **Sun Liguo**, Sun Jianguo, Robust disturbance rejection control design for integrated turbo-shaft engine/helicopter system, Chinese Journal of Aeronautics, 2010, vol. 5, written in Chinese and an English abstract available online

ACKNOWLEDGEMENTS

This thesis is the result of four years of research within the Division of Control and Simulation at the Aerospace Engineering Faculty in Delft University of Technology. Finishing this research project would not have been possible without the support of many people. I am very grateful to all of these people, but I would like to mention some of them in particular.

First of all, I would like to thank my promoter Professor Max Mulder for his strategic guidance, and all his other support. Thanks for giving me the opportunity and the freedom to pursue my PhD project.

I would like to thank my co-promoter Dr Ping Chu for his scientific guidance and insights that lead to many innovations during my research project: without Dr Chu's supervision, I could not have overcome the many research difficulties that I suffered from.

I would like to express my sincere gratitude to my daily supervisor Dr Coen de Visser for all our impressive discussions and his comments on many journal papers. I greatly appreciated our numerous discussions and our cooperation when writing journal papers, and I am looking forward to continuing our cooperation.

My gratitude also goes out to professor J.A. (Bob) Mulder for recruiting me into the department of Control and Operation, and for giving me a lot of helpful feedback on several journal papers. I can still clearly remember our first meeting in the faculty together with Dr Ping Chu shortly after I first arrived at the Netherlands.

I would like to thank my former supervisor from my Master's stage, Professor Jian-Guo Sun, for his unconditional support.

I would also like to thank Laurens van Eykeren for his kindness and patient help with scientific research and for translating my thesis summary from English to Dutch. I will never forget that you provided me with the chance to fly in Belgium under your supervision, and we visited Ghent and Antwerp together from the air. I also will remember that during the 2013th AIAA conference in Boston, USA, you introduced me to many famous professors in the field of fault detection and diagnosis and adaptive flight control. My thanks go to my office mates during my first year, Eddy van Oort and Wouter Falkena, for giving me unlimited help on both how to start my PhD research and on how to adapt to a new environment in the Netherlands. My gratitude goes to Thomas Lombaerts for giving me many helpful comments when reviewing papers I wrote during my research.

I would like to thank my tennis coach Lito (Dutch), and other buddies Hui Yu, Ping Liu, Jia Wan, Hao Chen, and Richard Huizenga, for all the refreshing exercise on the tennis court. A tennis game always gives me energy again after a day of hard research work.

My thanks also goes out to all the colleagues at the division of control and simulation; Olaf Stroosma, Guido de Croon, Marilena Pavel, Joost Ellerbroek, Laurens van Eykeren,

Wouter Falkena, Jan Comans, Bruno Correia Gracio, Deniz Yilmaz, Daan Pool, Rita Valente Pais, Herman Damveld, Paul de Jong, Arjen de Leege, Xander in't Veld, Herman Koolstra, Erik-Jan van Kampen, Hui Yu, Ferdinand Postema, Andries Muis, Harold Thung, Clark Borst, Hans Mulder, Elwin de Weerdt, Joost Venrooij, Peter Zaal, Eddy van Oort, Jia Wan, Peng Lu, Hann-woei Ho, Yazdi Jenie, Joao Aguiar Vieira Caetano, Jaime Junell, Rolf Klomp, Dyah Jatiningrum, Tommaso Mannucci, Ye Zhou, Emmanuel Sunil, Tao Lu and Sophie Armanini. I would like to thank you for all for the high and low times in the coffee corner, on the basketball court, and in many other places. I would also like to thank Bertine Markus, the secretary of the Control&Simulation group, for all her help. Where would we be without secretaries?

My gratitude goes out to my friends in Delft, Dirk Goedbloed, Ning Sun, Henk van Koppen, Na Zhu, and Morgan van Vliet (Canada). I greatly enjoyed our discussions about life and the meaning of life. You helped me get used to living in Delft and invited me to celebrate Christmas eves and New Year's eves with your families. I would also like to thank my long time friends Hui Yu, Jia Wan, Wen Sang, Ye-bo Li, Hai-bo Zhang, Bo Li, Hai-qiang Wang, Ping Liu, Hao Cui, Wei Meng, Lu Wang, Xiao-yu Zhang, Jing Chu, Ming Liu, Xu Jiang, Tao Xu and my past and current room mates Ya-qing Shu, Lan Song, Hua-yang Cai, Ke-fei Li, Peng Xu, Yun-long Gong, and Zhi-yong Wang, for providing both fresh perspectives on my research and some much needed social distraction. Particularly, I would like to thank Wen Sang for providing me with accomodation in and guiding me around Boston. I really enjoyed the days spent in MIT when I was attending conferences in the USA in 2012 and 2013.

My thanks also go to Miranda Aldham-Breary first for reading my thesis and providing feedback, and for helping me to improve my oral english. I would also like to thank my teacher from the spoken English course, Claire Taylor.

Then, I would like to thank my parents in law, Quan-gong Li and Ri-ting Gao, for all their kind support, interest and feedback in the past years.

I also want to express my gratitude to my parents, Yue Sun and Gui-Yin You. Thank you for your unconditional support, both materially and spiritually, constructive feedback, and care. Our endless discussions on any topic have been, and always will be, a great source of inspiration and guidance for me. I would also like to thank my sister, Li-yan Sun, for supporting me.

My gratitude goes to my beloved wife, Hong-juan Li. Without you I would not have been able to finish this thesis. With your support, love, enthusiasm, optimism, and determination, I gained the courage to face and overcome numerous difficulties. I see a bright future for our life together in which nothing is impossible.

Finally, I would like to thank the China Scholarship Council (CSC) for their four-year funding support, without which I could not have accomplished my Ph.D. study in the Netherlands.

Li Guo SUN
Delft, September 2014

LIST OF PUBLICATIONS

Journals:

1. **L.G. Sun**, C.C. de Visser, Q.P. Chu, J.A. Mulder, Online Aerodynamic Model Identification using a Recursive Sequential Method for Multivariate Splines, Journal of Guidance, Control, and Dynamics, Vol.36, No. 5, pp. 1278-1288, 2013 (doi:10.2514/1.60375)

2. **L.G. Sun**, C.C. de Visser, Q.P. Chu, W. Falkena, Hybrid Sensor-Based Backstepping Control Approach with its Application to Fault-Tolerant Flight Control, Journal of Guidance, Control, and Dynamics, Vol.37, No. 1, pp. 59-71, 2014 (doi:10.2514/1.61890)

3. **L.G. Sun**, C.C. de Visser, Q.P. Chu, J.A. Mulder, A novel adaptive kernel method with kernel centers determined by a support vector regression approach, Neurocomputing, Vol. 124, pp.111-119, 2014 (doi:10.1016/j.neucom.2013.07.023)

4. **L.G. Sun**, C.C. de Visser, Q.P. Chu, J.A. Mulder, Joint Sensor Based Backstepping For Fault-Tolerant Flight Control, Journal of Guidance, Control, and Dynamics, Accepted, 2014 (doi:10.2514/1.G000432)

5. **L.G. Sun**, C.C. de Visser, Q.P. Chu, Improving Model Structure Selection for Aerodynamic Model Identification with Multivariate Simplex B-splines, Journal of Guidance, Control, and Dynamics, Submitted, 2014

6. H.J. Tol, C.C. de Visser, **L.G. Sun**, E. van Kampen, Q.P. Chu, Nonlinear Multivariate Spline Based Adaptive Control Allocation with Online Aerodynamic Model Identification, Journal of Guidance, Control, and Dynamics, Submitted, 2014

Conference Proceedings:

1. **Invited Speaker:** L.G. Sun, C.C. de Visser,Q.P. Chu, Fault-tolerant Control for a Civil Transportation Aircraft, 5th EUCASS conference, **Invited session**, Munich, Germany, July, 2013

2. L.G. Sun, C.C. de Visser and Q.P. Chu, A Recursive Sequential Method for Multivariate Splines Towards Online Aerodynamic Model Identification, AIAA Guidance, Navigation, and Control conference, Mineapolis, Minesota, August, 2012

3. L.G. Sun, C.C. de Visser, Q.P. Chu, A New Substitution Based Recursive B-Splines Method for Aerodynamic Model Identification, 2nd CEAS EUROGNC13, pp: 858-871, Delft, The Netherlands, April, 2013

4. L.G. Sun, C.C. de Visser, Q.P. Chu, A New Joint Sensor Based Backstepping Approach for Fault-tolerant Flight Control, 2nd CEAS EUROGNC13,pp: 804-823, Delft, The Netherlands, April, 2013

5. L.G. Sun, C.C. de Visser, Q.P. Chu and W. Falkena, A Hybrid Sensor Based Backstepping Control Approach with its Application to Fault-Tolerant Flight Control, AIAA Guidance, Navigation, and Control conference, Boston, MA, August, 2013

6. L.G. Sun, C.C. de Visser, Q.P. Chu and W. Falkena, A Joint Sensor Based Backstepping Approach For Fault-Tolerant Flight Control of a Large Civil Aircraft, AIAA Guidance, Navigation, and Control conference, Boston, MA, August, 2013

7. L.G. Sun, C.C. de Visser, Q.P. Chu, A new Multivariate Tensor-Product Simplex B-splines with its Application in Real-Time Modular Adaptive Flight Control, AIAA Guidance, Navigation, and Control conference (SCITECH), National Harbor, Maryland, January, 2014

Book Chapters:

1. L.G. Sun, C.C. de Visser, Q.P. Chu, A New Substitution Based Recursive B-Splines Method for Aerodynamic Model Identification, Advances in Aerospace Guidance, Navigation and Control: Selected Papers of the Second CEAS Specialist Conference on Guidance, Navigation and Control,ISBN:3642382525, 9783642382529, pp. 233-245, Springer London, 2013

Acadamic Activity:

1. Reviewer for Automatica, IF 2.919, since 2013
2. Reviewer for Information Sciences, IF 3.643, since 2014
3. Reviewer for Mathematical Problems in Engineering, IF 1.383, since 2013
4. Reviewer for IEEE Transactions on Signal Processing, IF 2.813, since 2013
5. Reviewer for Neurocomputing (Journal), IF 2.005, since 2014